PERGAMON

Transportation Research Part B 37 (2003) 541–560

TRANSPORTATION RESEARCH PART B

www.elsevier.com/locate/trb

Berth scheduling by simulated annealing

Kap Hwan Kim *, Kyung Chan Moon [1]

Department of Industrial Engineering, Pusan National University, Changjeon-dong, Kumjeong-ku, Busan 609-735, South Korea

Received 21 March 2001; received in revised form 18 March 2002; accepted 28 March 2002

Abstract

The objective of the berth-scheduling problem is to determine the berthing times and positions of containerships in port container terminals. Every vessel requires a specific amount of space in a wharf for a predetermined length of time to unload and load containers. In this study, a mixed-integer-linear-programming (MIP) model was formulated for the berth-scheduling problem. The simulated annealing algorithm was applied to the berth-scheduling problem to find near-optimal solutions. Experimental results showed that the simulated annealing algorithm obtains solutions that are similar to the optimal solutions found by the MIP model.

Keywords: Berth scheduling; Container terminals; Mixed-integer-programming; Simulated annealing

1. Introduction

Managers in many container terminals attempt to reduce costs by efficiently utilizing resources, including human resources, berths, container yards, container cranes, and various yard equipment. Among all the resources, berths are the most important resource and good schedules of berths improve customers' satisfaction and increase port throughput, leading to higher revenues of port. Port managers usually schedule the usage of berths by an intuitive trial-and-error method supported by a schedule board or a graphic-user-interface in a computer system. This paper attempts to maximize the utilization of a wharf and to satisfy various constraints for berthing container vessels by using an analytical approach.

* Corresponding author. Tel.: +82-51-510-2419; fax: +82-51-512-7603.
E-mail addresses: kapkim@pusan.ac.kr (K.H. Kim), kcmoon@pusan.ac.kr (K.C. Moon).
[1] Tel.: +82-51-510-1483; fax: +82-51-512-7603.

0191-2615/03/$ - see front matter
doi:10.1016/S0191-2615(02)00027-9

When scheduling berth usage, the berthing time and position of every vessel must be determined. In the process, several factors must be considered, including the length of each vessel, the arrival time of each vessel, the number of containers for discharging and loading, and the storage location of outbound containers to be loaded onto the corresponding vessel.

Fig. 1 illustrates an example of the berth schedule in a port container terminal. The horizontal axis represents the position on a wharf, while the vertical one is the time axis. Each rectangle represents the schedule of the corresponding vessel. The horizontal position of a rectangle for a vessel corresponds to the berthing position of the vessel, while the vertical position of the rectangle indicates the time of the ship operation for the corresponding vessel.

The objective of berth scheduling is to minimize the penalty cost resulting from delays in the departures of vessels and the additional handling costs resulting from non-optimal locations of vessels in a wharf. Carriers usually inform a terminal operator the expected arrival time and the requested departure time of vessels. Based on the information, the terminal operator tries to meet the requested departure time of all vessels. However, when the arrival rate of vessels is high or when unexpected arrivals occur, it may not be possible to complete services for all the vessels by the requested departure times. Thus, the departures of some vessels may be delayed. Also, terminal operators usually have different priorities for different types of vessels. The priorities can be considered by converting them into cost coefficients of the penalty cost for vessels in the objective function.

The berthing position is also an important decision variable for the following reasons. Containers for a vessel usually start to arrive at a yard from several days before the vessel arrives at a port. Thus, if a vessel is berthed at the location near to the storage location of containers to be loaded onto the vessel, the delivery cost for the containers by yard trucks or straddle carriers can

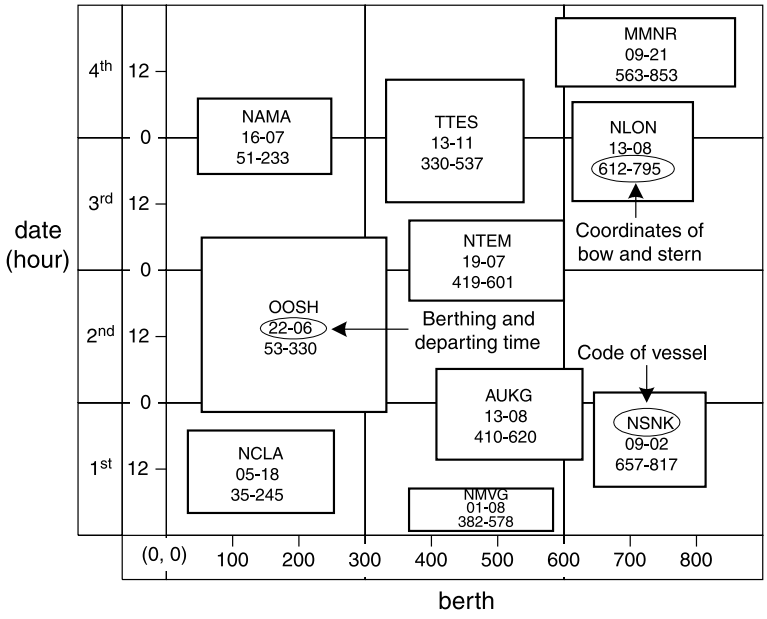


Fig. 1. An example of a berth schedule.

be minimized. In the case, the least-cost berthing location of the vessel is the position on the wharf on which the center of the vessel coincides with the average coordinate of outbound containers that already arrived at the yard and will be loaded onto the vessel. Also, some berthing locations may be preferable over other locations because of factors such as long-term contracts that specify the usage of berths with carriers, the minimum draft required for a vessel, different levels of waves according to locations in a wharf, etc.

Numerous studies have been conducted regarding the improvement of the efficiency of ship operations in port container terminals. These studies include the load-sequencing problem by Cho (1982), Cojeen and Dyke (1976), Gifford (1981), and Kim and Kim (1999), and the crane-scheduling problem, by Daganzo (1989) and Perterkofsky and Daganzo (1990). Brown et al. (1994), Imai et al. (2001), Lai and Shih (1992), Li et al. (1998), and Lim (1998) dealt with the berth-scheduling problem, the topic of this paper.

Through a simulation experiment, Lai and Shih (1992) compared three berth-allocation rules for container vessels. They assumed that a wharf, which can be represented as a continuous line, can be partitioned into several sections, to each of which only one vessel can be allocated at a specific time. The berth-allocation rules considered the following factors: the available sections of a wharf, the expected completion time of a vessel at each available section, and size and arrival time of each vessel.

By considering various practical constraints, Brown et al. (1994) formulated an integer-programming model for assigning available sections of a wharf to vessels. They assumed a wharf to be a collection of discrete berthing sections. Lim (1998) first considered a wharf to be a continuous space rather than a collection of partitioned sections, and the berth-scheduling problem was formulated as a two-dimensional packing problem. However, he assumed that the berthing times of all vessels were fixed. He proposed a heuristic method for minimizing the total length of a wharf occupied by vessels.

Li et al. (1998) formulated the berth-scheduling problem as a scheduling problem with a single processor through which multiple jobs can be processed simultaneously. The problem assumed that all vessels had already arrived, and the minimization of the make-span was attempted based on that assumption. They suggested a first-fit-decreasing heuristic rule for which a numerical experiment was conducted. They also addressed the case where relocations of vessels are allowed. However, they did not suggest a method to obtain optimal solutions.

Imai et al. (2001) also assumed a wharf to be a collection of discrete berthing sections. They attempted to minimize the waiting time of vessels and provided a mixed-integer-programming model for allocating berthing sections to vessels. They also provided a heuristic procedure based on the Lagrangean relaxation of the original problem.

The study described in this paper is unique in the following three aspects: First, unlike in Lim's study (1998), this study attempts to simultaneously determine the berthing time and the berthing location. Second, unlike in the studies by Brown et al. (1994), Imai et al. (2001), and Lai and Shih (1992), the wharf is considered to be a continuous space rather than a collection of partitioned sections. Third, unlike in the studies by Lai and Shih (1992) and Li et al. (1998) which suggested heuristic rules, this study suggests an optimizing method.

In the area of facility layouts, a useful formulation similar to ours can be found in Heragu and Kusiak (1991) and Tompkins et al. (1996). The formulations for layout attempts to locate small rectangles, which represent departments, within a large rectangle, which represents a

building. The objective of the facility layout problem is to minimize the sum of material flows multiplied by the distance between corresponding rectangles. The two-dimensional cutting-stock problem is also similar to the berth-scheduling problem. However, the objective of the cutting-stock problem is to minimize the amount of scrap materials and thus the positions of cutouts on a plane—which are important in the berth-scheduling problem—are not related to the objective function.

The single-machine scheduling problem for minimizing the total earliness and tardiness penalty (Fry et al., 1987) is a special case of the berth-scheduling problem. Because the former problem determines only the starting times of jobs, it is a single-dimensional version of the berth-scheduling problem, in which the berthing positions as well as the berthing times must be determined.

The next section presents a mathematical formulation of the berth-scheduling problem. Section 3 introduces several properties of the optimal solution. Section 4 proposes a heuristic, and a simulated annealing algorithm for the berth-scheduling problem. Finally, Section 5 presents concluding remarks.

2. Problem formulation

This section presents a mathematical formulation for the berth-scheduling problem. The following notations are used for the formulation:

N The total number of vessels.

L The length of a wharf.

p_i The lowest-cost berthing location of vessel i. This location is represented by the x coordinate of the left-most end of the vessel, and the location is determined by considering the distribution of previously arrived containers or the designated location for a specific vessel. The reference point for the x coordinate is the left-most boundary of the wharf.

x_i The berthing position of vessel i (a decision variable).

y_i The berthing time of vessel i (a decision variable).

a_i The estimated arrival time of vessel i.

d_i The departure time requested by vessel i.

b_i The time required for the ship operation for vessel i. This value includes the time allowance between the departure of a vessel and the berthing of another vessel.

c_{1i} The additional travel cost per unit distance for delivering the containers for vessel i, resulting from non-optimal berthing location relative to the lowest-cost point.

c_{2i} The penalty cost per unit time of vessel i, resulting from a departure delayed beyond the requested due time. When the departure of a vessel is delayed behind her schedule, she must speed up her voyage to catch up with her schedule at the next port, which results in an extra fuel consumption. The penalty cost c_{2i} may be estimated from the expected extra fuel consumption.

l_i The length of vessel i. This value includes the requested gap between adjacent vessels.

z_{ij}^x $\begin{cases} 1 : \text{if vessel } i \text{ is located to the left of vessel } j \text{ on the wharf}, \\ 0 : \text{otherwise}. \end{cases}$

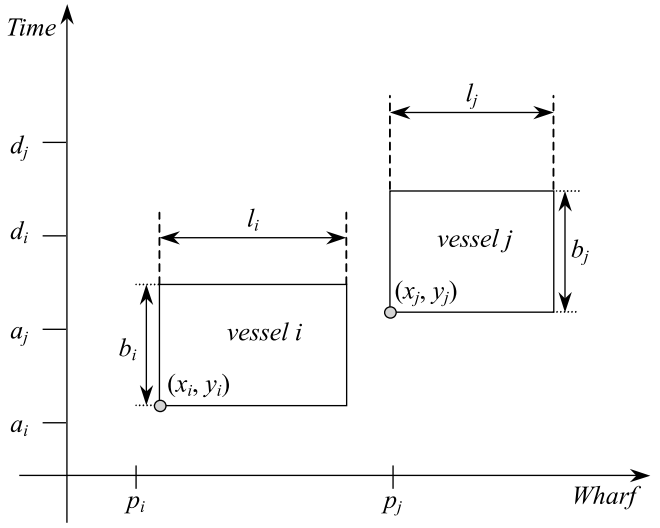


Fig. 2. An illustration of notations for a berth-schedule on the wharf-time space.

$$z_{ij}^{y} \quad \begin{cases} 1 : \text{if vessel } i \text{ is berthed before vessel } j \text{ in time,} \\ 0 : \text{otherwise.} \end{cases}$$

Fig. 2 illustrates the wharf-time space and several notations. Then, the objective function of the berth-scheduling problem can be written as follows:

$$\text{Min} \sum_{i=1}^{N} \{c_{1i}|x_i - p_i| + c_{2i}(y_i + b_i - d_i)^+\}, \tag{1}$$

where $x^+ = \max\{0, x\}$.

The first term of the objective function represents the cost that results from the non-optimal berthing location, and the second term is the penalty cost resulting from the delay in the departure of vessels. Let $|x_i - p_i|$ be α_i^+ when $x_i - p_i \geqslant 0$, and α_i^- when $x_i - p_i < 0$. Also, let $(y_i + b_i - d_i)$ be β_i^+ when $y_i + b_i - d_i \geqslant 0$, and β_i^- otherwise. Then, the berth-scheduling problem can be formulated as follows:

$$\text{Min} \sum_{i=1}^{N} \{c_{1i}(\alpha_i^+ + \alpha_i^-) + c_{2i}\beta_i^+\} \tag{2}$$

subject to

$$x_i - p_i = \alpha_i^+ - \alpha_i^- \quad \text{for all } i \tag{3}$$

$$y_i + b_i - d_i = \beta_i^+ - \beta_i^- \quad \text{for all } i \tag{4}$$

$$x_i + l_i \leqslant L \quad \text{for all } i \tag{5}$$

$$x_i + l_i \leqslant x_j + M(1 - z_{ij}^x) \quad \text{for all } i \text{ and } j,\ i \neq j, \text{ and a large positive number } M \tag{6}$$

$$y_i + b_i \leqslant y_j + M(1 - z_{ij}^y) \quad \text{for all } i \text{ and } j,\ i \neq j, \text{ and a large positive number } M \tag{7}$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geqslant 1 \quad \text{for all } i \text{ and } j,\ i \neq j \tag{8}$$

$$y_i \geqslant a_i \quad \text{for all } i \tag{9}$$

$$\alpha_i^+, \alpha_i^-, \beta_i^+, \beta_i^-, x_i \geqslant 0 \quad \text{for all } i \tag{10}$$

$$z_{ij}^x, z_{ij}^y \ 0/1 \text{ integer} \quad \text{for all } i \text{ and } j,\ i \neq j \tag{11}$$

Constraints (3) and (4) are related to the definitions of $\alpha_i^+, \alpha_i^-, \beta_i^+$, and β_i^-. Constraint (5) implies that the position of the rightmost end of vessel i is restricted by the length of the wharf. Constraint (6) or (7) is effective only when z_{ij}^x or z_{ij}^y equals one. Constraint (8) excludes the case where two vessels are in conflict with each other with respect to the berthing time and the berthing position. That is, constraint (8) excludes the case $z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y = 0$, in which case the rectangles representing the schedules for vessel i and j overlap. Constraint (9) implies that a vessel cannot berth before she arrives at a port.

A simplified version of the model described in this study, in which the berthing times of all the vessels are fixed, is known to be NP-hard (Lim, 1998). Because the computational time to solve formulation (2)–(11) with more than seven vessels by using a commercial software (LINDO®) exceeded a reasonable limit and the number of vessels in practical problems was about 20, a simulated annealing algorithm is suggested in Sections 3 and 4.

3. Properties of the optimal solution

In this paper, solutions are encoded by a sequence of vessels to apply the simulated annealing algorithm to the berth-scheduling problem. To decode a sequence of vessels into a solution, it must be possible to obtain near-optimal positions of rectangles of vessels in the time-wharf plane from the given sequence of vessels. This section discusses several useful properties of the optimal solution.

The following introduces several definitions necessary for explaining the concept of stability.

Definition 1. An x-cluster is defined as a set of rectangles (vessels) whose vertical sides are in contact with each other. Also, a y-cluster is defined as a set of rectangles (vessels) whose horizontal sides are in contact with each other.

Definition 2. A cluster of vessels is said to be "stable" if the total cost for vessels in the cluster cannot be reduced by moving the cluster in the positive or negative direction.

Property 1. *Clusters in the optimal solution to formulations* (2)–(11) *are stable.*

Proof. This property holds by the definition of the stability of clusters. □

Property 2. *When an x-cluster is stable, either of the following two conditions holds*: (1) *one or more rectangles in the x-cluster are located at their least-cost points on the x-axis*; (2) *a side of at least one rectangle in the cluster is on the right or the left boundary of the wharf.*

Proof. Let the leftmost rectangle in a cluster be called a "reference rectangle" of the cluster. Let x be the horizontal coordinate of the left side of the reference rectangle of the cluster. Also, let r_i be the relative horizontal coordinate of the left side of rectangle i from the left side of the reference rectangle of the associated cluster. Then, the cost of the ith rectangle can be expressed as $c_{1i}|x + r_i - p_i|$. Thus, the total cost can be expressed by $\Sigma_{i\in Ck} c_{1i}|x + r_i - p_i|$, where c_k is a set of rectangles in the kth x-cluster. This problem is equivalent to the minisum problem of a single facility under the rectilinear distance measure. In the expression of the total cost, $(p_i - r_i)$ can be considered to be the location of the ith existing facility. That is, x^* can be found by applying the property of the median condition (Francis et al., 1992). In this case, one or more rectangles must be located at their least-cost locations for the cluster to be stable. However, when a rectangle in the cluster is in contact with either of the two boundaries of the wharf, no rectangle may be located at its least-cost position. Thus, the conclusion holds. □

Property 3. *When a y-cluster is stable, either of the following two conditions holds*: (1) *the berthing times of all the vessels in the y-cluster are located in the range of* $[a_i, d_i - b_i]$; (2) *the berthing time of one or more vessels in the cluster is at its arrival time* (a_i) *or zero*.

Proof. Similar to the proof for Property 2. □

When a layout of rectangles of vessels is provided for a given sequence of vessels, we can improve the layout by checking the stability of clusters and stabilizing unstable clusters, which can be done by just moving clusters into such a direction that will reduce the cost. According to Properties 2 and 3, in the process, it is sufficient to consider only several discrete points on x-axis or y-axis as candidate positions for a cluster. This improvement process will be used to decode a solution in the next section.

4. Applying the simulated annealing method

To describe the simulated annealing algorithm, an encoding and decoding method of solutions must be defined first.

4.1. Encoding and decoding of solutions

A solution is encoded by a sequence of vessels. To decode a sequence of vessels into a solution, rectangles of vessels will be added into the time-wharf plane one by one. For a given layout of rectangles in the time-wharf plane, a right/upper corner, (s_i, t_i), of rectangle is said to be dominated if there exists another corner, (s_j, t_j), such that that $s_i \leqslant s_j$ and $t_i \leqslant t_j$ and either of the two inequalities strictly holds. By using the set of non-dominated right/upper corners, a downward stairway can be constructed as illustrated by bold lines in Fig. 3(ii). The line segments between adjacent breaking points and ending points will be called "profile segments" in the following description. The following provides an overall heuristic procedure for obtaining a berth schedule for a given sequence of vessels.

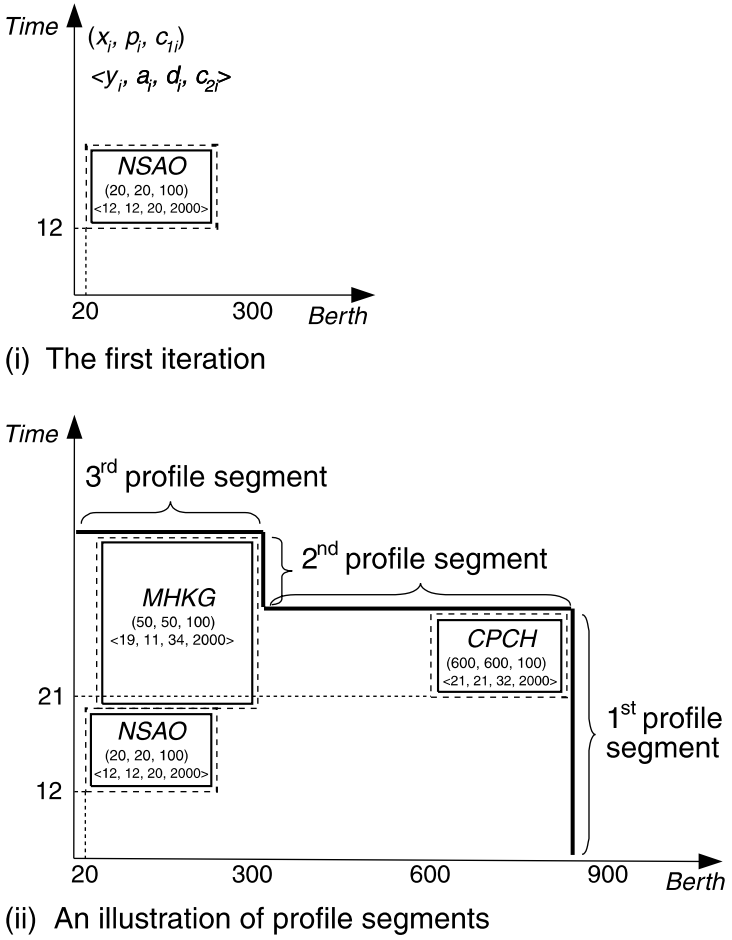


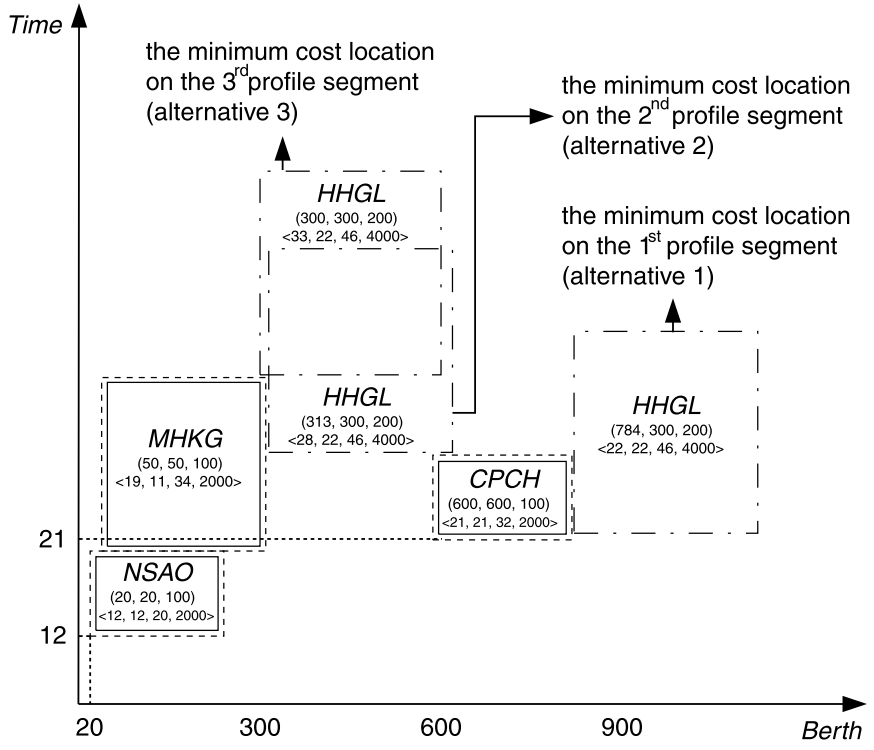
Fig. 3. An illustration of the heuristic algorithm.

Step 0: $i = 1$. Locate the first rectangle in the sequence at the lowest-cost point.

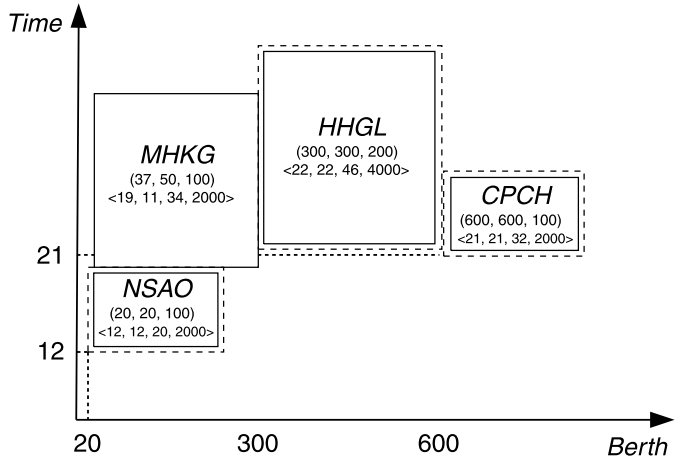
Step 1: $i = i + 1$. If $i > N$, stop. Otherwise, construct profile segments from the current layout of rectangles. Let $j = 0$.

Step 2: $j = j + 1$. If $j > m + 1$, go to step 1. Otherwise, locate (x_i, y_i) at the lowest-cost point on the jth profile segment. Identify x- and y-clusters that the ith rectangle is associated with. Check the stability of the associated clusters by the procedure in Appendix B. Include unstable clusters into the set of unstable clusters. Go to step 3.

Step 3: Check if there remains an unstable x-cluster. If none exists, go to step 4. Otherwise, select an x-cluster that is unstable. Apply the procedure in Appendix C for stabilizing x-cluster to the selected x-cluster. In the process, one or more unstable y-clusters may be created. Include such unstable y-clusters into the set of unstable clusters. Go to the beginning of this step.



(iii) The minimum cost location of HHGL on each segment



(iv) Alternative 2 after stabilization

Fig. 3 (*continued*)

Step 4: Check if there remains an unstable y-cluster. If none exists, check if there remains an unstable x-cluster. If there remains no unstable x-cluster, go to step 2. If there remains an

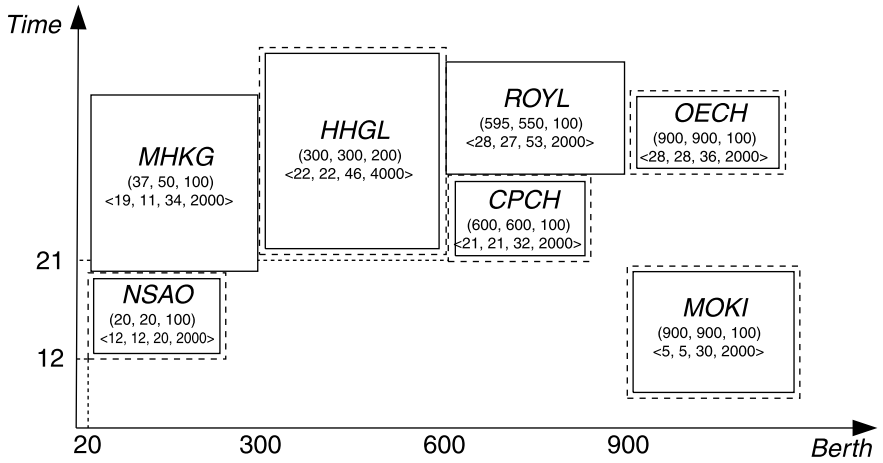


Fig. 3 (*continued*)

unstable x-cluster, go to step 3. If there still remains an unstable y-cluster, select an unstable y-cluster. Apply the procedure for stabilizing y-cluster in Appendix C to the selected y-cluster. In the process, one or more unstable x-clusters may be created. Include such x-clusters into the set of unstable clusters. Go to the beginning of this step.

The procedure for checking the stability of clusters, which is described in Appendix B, uses the cluster graph that can be constructed following the procedure in Appendix A. The procedure for stabilizing a cluster, which is described in Appendix C, is basically to push the cluster to the direction of reducing the total cost of the cluster until the cost does not decrease further. By Properties 2 and 3, when pushing a cluster, it is sufficient to check only a finite number of positions to find the stopping position.

At the beginning of each iteration in the above procedure, a new vessel is added to the set of vessels already positioned. The initial position of the new vessel is the outside of the stairway that is constructed by a non-dominated set of right/upper corners of already positioned rectangles. If x-cluster is unstable, then the procedure for stabilizing x-cluster in the left-hand direction is applied. And if y-cluster is unstable, the procedure for stabilizing y-cluster in the downward direction is applied. Therefore, it is expected that first locating a vessel with a small p_i and d_i first will result in a good performance, compared with other sequencing rules.

The heuristic algorithm is illustrated by using the data shown in Table 1. The penalty cost per hour was set to be \$4000 for HHGL and \$2000 for the other vessels. The additional handling cost incurred by non-optimal berthing locations was set to be \$200 per meter for HHGL and \$100 for the other vessels. Let the sequence of vessels be NSAO, MHKG, CPCH, HHGL, MOKI, OECH, and ROYL.

Fig. 3 shows an illustration of the heuristic algorithm. Note that the rectangles formed by solid lines represent real locations, while those formed by dotted lines represent the lowest-cost points. The first vessel, NSAQ, is located at its lowest-cost point, as shown in Fig. 3(i). Suppose that the

Table 1
Input data for an illustrative example

Vessels	l_i (m)	b_i (h)	a_i	d_i	p_i (m)
NSAO	182	7	12	20	20
HHGL	295	22	22	46	300
ROYL	294	13	27	53	550
OECH	185	6	28	36	900
MHKG	263	14	11	34	50
CPCH	184	7	21	32	600
MOKI	262	18	5	30	900

positions of rectangles for NSAO, MHKG, and CPCH are determined. In the following, the procedure of locating the rectangle for HHGL—which is the next vessel to be positioned—is illustrated. Based on the rectangles for NSAO, MHKG, and CPCH whose locations were stabilized in the previous iteration, the profile of the stairway is drawn and partitioned into three segments, as shown in Fig. 3(ii). Fig. 3(iii) shows the initial location of HHGL on each profile segment before stabilization. Fig. 3(iv) shows alternative 2, which is obtained by positioning HHGL on profile segment 2 and stabilizing the corresponding clusters. Three alternatives are obtained. Increases in cost resulting from additionally positioning HHGL on three profile segments were \$68,100, \$1300, and \$36,000, respectively. Thus, the solution in Fig. 3(iv) will be chosen as the final solution of the fourth iteration. After all the iterations are completed, the solution in Fig. 3(v) will be found to be the final solution.

4.2. Finding a near-optimal solution

In the simulated annealing method, the quality of the solution depends on the control parameters and the schedule of the temperature. In typical implementations, the simulated annealing approach involves a pair of nested loops and three additional parameters, a cooling rate, $0 < r < 1$, and a temperature length, R (see the algorithm below). The following describes the procedure for obtaining a berth schedule by using simulated annealing:

Step 1. Obtain an initial solution S. Let $T = T_0$.
Step 2. Repeat the following steps until one of the stopping conditions becomes true.
  Step 2.1. Perform the following loop R times.
    Step 2.1.1. Pick a random neighbor S' of S.
    Step 2.1.2. Let $\delta = \text{cost}(S') - \text{cost}(S)$. The cost is evaluated by Eq. (2).
    Step 2.1.3. If $\delta < 0$ (downhill move), set $S = S'$.
    Step 2.1.4. If $\delta \geqslant 0$ (uphill move), generate random number, x, from the interval, (0, 1); If $x < \exp(-\delta/T)$, then set $S = S'$.
  Step 2.2. Set $T = rT$ (Reduce the temperature).

The initial solution: It is expected that the better the initial solution, the better the final solution. The initial sequence of vessels is in the increasing order of $\alpha p_i/L + (1-\alpha)d_i/\max_i\{d_i\}$, where

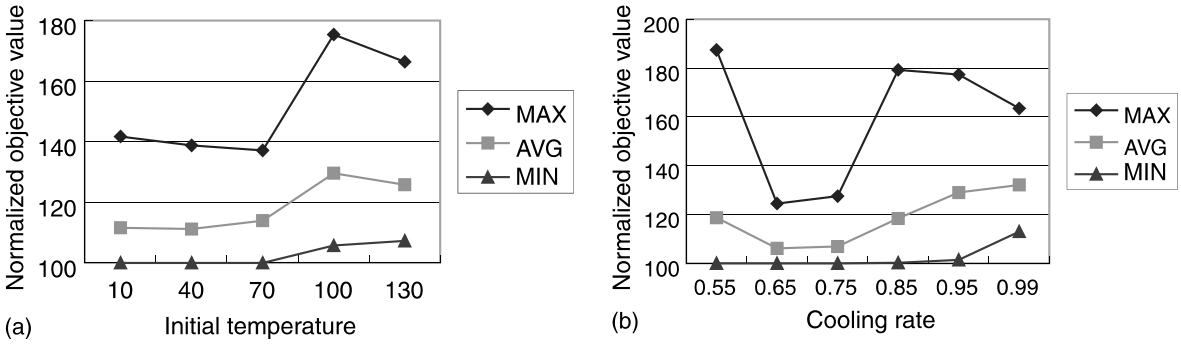


Fig. 4. Effects of initial temperature and cooling rate on objective values.

$0 < \alpha < 1$. A higher priority is given to a vessel with lower values of p_i and d_i because the decoding algorithm begins to locate vessels from the lower-left corner of the rectangle, which represents the berth-time space, to the upper-right direction.

Neighborhood: Pairwise exchanges (swapping the positions of two randomly selected elements) are used to obtain neighbors.

The initial temperature and the cooling rate: The selection of the initial temperature and the cooling rate influences the quality of the solutions obtained through the simulated annealing algorithm. In general, a slower schedule (i.e., one that starts from a higher temperature, T; has a larger cooling rate, r; and has a larger temperature length, R) will lead to better solutions. On the other hand, slower schedules tend to consume more computation time. This study attempted to select a set of cooling parameters that were expected to produce good, though not necessarily optimal, solutions within reasonable time. Five problems with 38–40 vessels were tested. The problems were solved with five different initial temperatures: 10, 40, 70, 100, and 130 °C. They were also solved with six different cooling rates: 0.55, 0.65, 0.75, 0.85, 0.95, and 0.99. For each combination of parameters, each problem was solved twice with different streams of random numbers. Thus, each problem was solved 48 times. Each objective value was normalized by calculating $100 \times$ (the objective value/the lowest objective value). The average, best, and worst normalized values were plotted for different initial temperatures and cooling rates, as shown in Fig. 4(i) and (ii), respectively. The minimum average objective values were found when the initial temperature was 40 and the cooling rate was 0.65; these values were used as the parameters in the succeeding experiments.

The length of temperature: The length of temperature was set to $N(N-1)/2$, which represents the size of a neighborhood of a solution.

The stopping criterion: When one of the following three conditions holds, the iterations are stopped:

1. A solution whose objective value is zero is found.
2. Temperature becomes less than 1.
3. The best value of the objective function has not been changed during the previous five consecutive external loops.

4.3. A numerical experiment

A numerical experiment was conducted to evaluate the performance of the simulated annealing algorithm described in this study. Real data for a berth and sizes of vessels were collected from the Pusan East Container Terminal (PECT), which is one of the largest container terminals in Korea. Data for vessels were collected from January to May in 1999. The length of the wharf at PECT is 1200 m. It was assumed that the safety allowance between vessels at the wharf is 20 m and at least 1 h is needed for the departure or the berthing of a vessel.

The arrival times and the requested departure times were generated randomly. However, the difference between the arrival time and the requested departure time for a vessel was maintained between 1.0 and 2.0 times the ship operation time. Also, p_i was also generated randomly within the range of the wharf. First, in order to evaluate the performance of the simulated annealing algorithm in this study, 20 problems were solved by LINDO® for the formulation (2)–(11) and by the simulated annealing algorithm, respectively. The scheduling horizon was set to be 72 h. The problem size was restricted because the computational time for LINDO® exceeded a reasonable limit when the number of vessels became larger than 7 and the scheduling horizon exceeded 72 h.

The cooling schedule was set to $T = 40$, $r = 0.65$, and $R = N(N-1)/2$. Cost coefficients c_{1i} and c_{2i} of the problems were set to be $\$10l_i/230$ and $\$2000l_i/230$, respectively. 20 problems were solved. Each problem was solved 5 times with different streams of random numbers. Table 2 shows the maximum objective values and the computational time for the 5 trials. Note that by simulated

Table 2
A comparison between the objective values generated by LINDO® and the simulated annealing algorithm

Problem number	LINDO®	Simulated annealing		
	Objective value	Computational time (s)	Objective value (maximum)	Computational time (s) (maximum)
1	0	10	0	<1[a]
2	0	18	0	<1
3	0	4	0	<1
4	1820	84	1820	<1
5	318	8	318	1
6	2662	6	2662	1
7	0	4	0	<1
8	912	6	912	1
9	1164	107	1164	1
10	636	61	636	1
11	2732	77	2732	1
12	99	9	99	1
13	640	10	640	1
14	0	7	0	<1
15	1244	141	1244	1
16	480	9	480	1
17	0	11	0	<1
18	789	31	1140	1
19	0	9	0	<1
20	2355	75	2794	<1

[a] The computational time was less than 1 second.

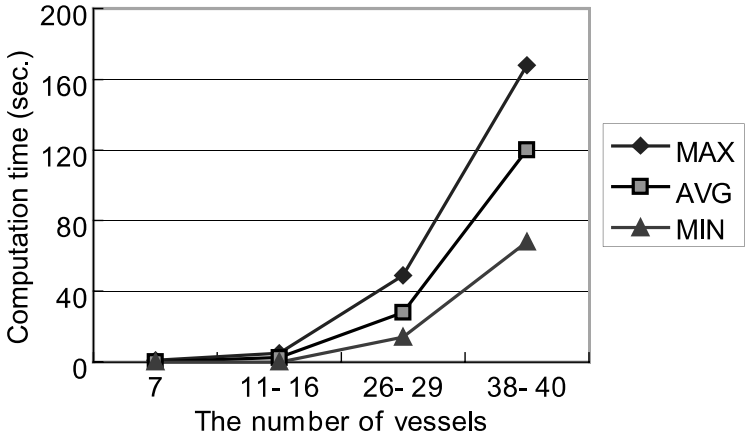


Fig. 5. Trend of the computation time for problems of different sizes.

annealing, the optimal solutions were obtained for 18 of the 20 example problems, which implies that the simulated annealing method is an effective method for solving the berth scheduling problems.

Three sets of problems were solved by the simulated annealing algorithm. Each set consisted of 20 problems with 11–16, 26–29, and 38–40 vessels, respectively. Fig. 5 shows the trend of the computation time for problems of different sizes. It was observed that the average and the range increase rapidly as the problem size increases. However, considering that terminal operators usually schedule vessels one week in advance and that the maximum number of vessels that arrive during one week at terminals of the size similar to PECT is usually less than 25, the computational time is satisfactory for practical uses. For each problem, the ratio of the range of objective values for five trials to the lowest objective value, which can be expressed by (the highest objective value during the five trials—the lowest objective value during the five trials)/(the lowest objective value during the five trials), was calculated. Fig. 6 shows the average ratio of the range for 20

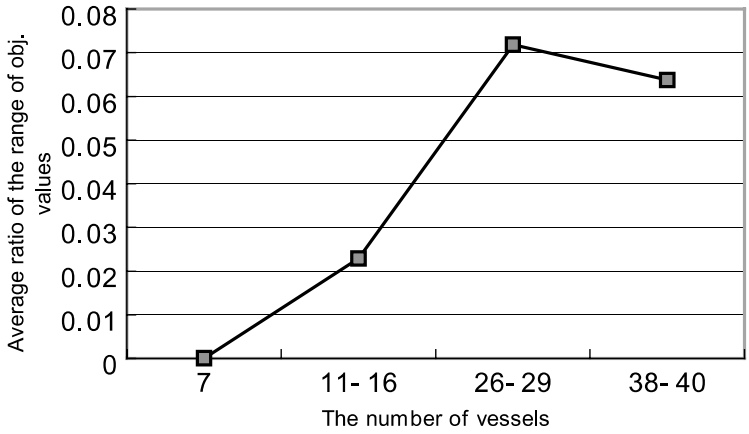


Fig. 6. Average ratio of the range of objective values to the lowest objective value for different problem sizes.

problems with different number of vessels. The average ratio tends to increase as the number of vessels increases. However, considering that the average ratio is less than 10%, we can conclude that the objective values by the simulated annealing method are consistent for different trials.

Fig. 7 shows the percentage of solutions that have objectives values exceeding the lowest values by a specified amount. According to Fig. 7, the percentage of solutions whose objective values exceed the lowest values by larger than 10% is less than 5%. This result implies that the solutions from different trials of the simulated annealing algorithm are consistent in their objective values.

In the problem of this paper, different vessels compete for positions of the cost as low as possible with each other. If the lowest positions of vessels do not overlap with each other, then it is expected that the solutions can be easily obtained. However, if the overlapped area of rectangles of different vessels is large, then it is expected that the computational time and the objective value become large, which could be observed in Figs. 8 and 9 drawn from the results of problems with 26–29 vessels. Fig. 8 shows that the computational time increases as the ratio of the overlapped area of rectangles of vessels—when they are located at their least cost positions—to the total area of rectangles increases. Also, Fig. 9 shows that the objective value increases as the ratio of the overlapped area increases.

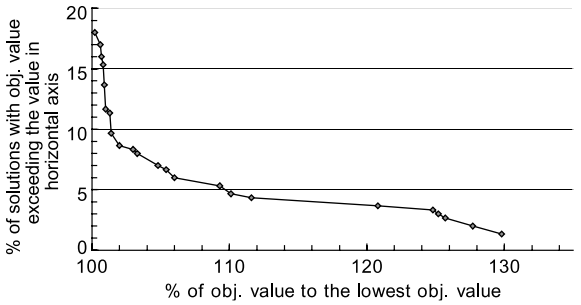


Fig. 7. The percentage of objective values greater than the lowest by a specified percentage.

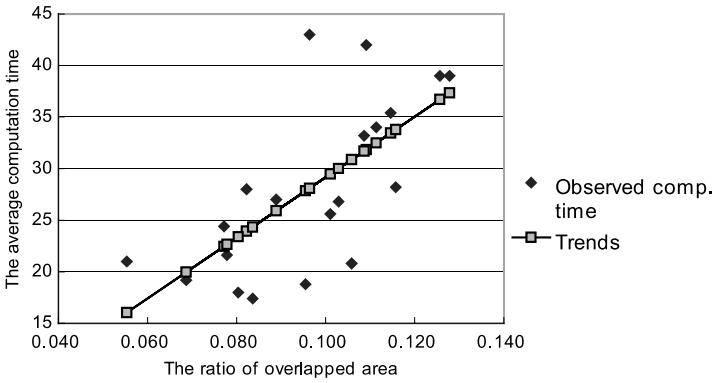


Fig. 8. Trends of the average computational time with respect to the ratio of overlapped area.

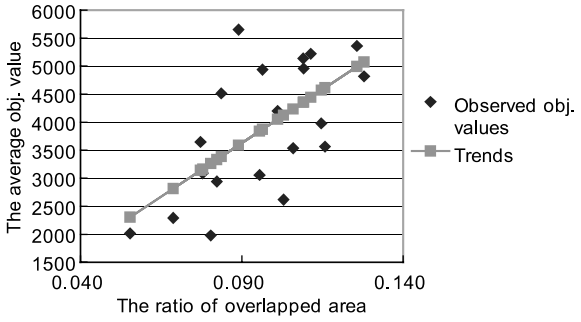


Fig. 9. Trends of the average objective value with respect to the ratio of overlapped area.

5. Conclusions

A linear integer program was formulated for the berth-scheduling problem and the problem was solved using the LINDO® package. The computational time of LINDO® increased rapidly when the number of vessels became higher than 7 and the length of the scheduling horizon exceeded 72 h. Thus, it was concluded that it is impractical to directly solve the linear integer model, and that a heuristic algorithm is necessary. Some properties of the optimal solutions were investigated. Based on the properties, the simulated annealing algorithm for the berth-scheduling problem was suggested. The performance of the simulated annealing algorithm was compared with that of the optimizing technique. It was found that the simulated annealing algorithm results in near-optimal solutions and that the computational time was within the limits of practical usage.

A numerical experiment showed that the computational time and the quality of solutions depend on the number of vessels and the ratio of the overlapped area of rectangles—when they are positioned at their least cost locations—to the total area of rectangles.

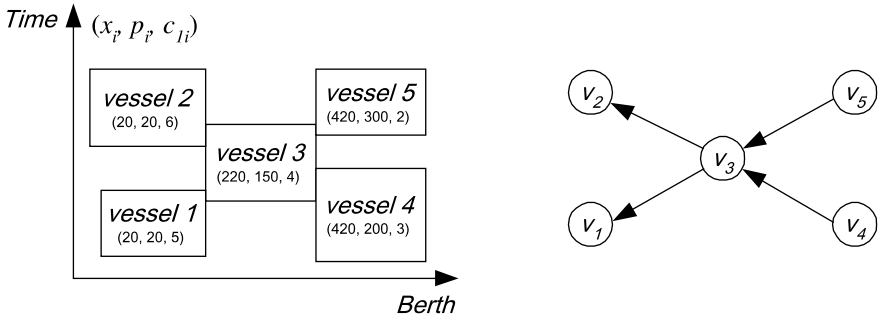
In practice, many planners simultaneously consider the schedule of quay cranes and that of the berth. That is, the berthing time may vary according to the number of cranes assigned to a corresponding vessel. Although the issue of crane scheduling was not incorporated into this model, it is a promising subject for further study. This study assumed a wharf in the shape of a straight line. However, there are practical examples of wharfs in different shapes or there may be multiple wharfs in the same area. The berth-scheduling problems in these cases are also open to future studies.

Acknowledgement

This research has been supported in part by Brain Korea 21 Program (1999–2002).

Appendix A. A graphical representation of clusters

It is assumed that, for any two disjoint subsets of vessels (V_1 and V_2), the sum of l_i for all $i \in V_1$ is not equal to the sum of l_j for all $j \in V_2$. It is also assumed that, for any two disjoint subsets of



(i) A berth schedule (ii) Graphic representation of the berth schedule

Fig. 10. An illustration of an x-cluster and its cluster graph.

vessels (V_1 and V_2), the sum of b_i for all $i \in V_1$ is not equal to that for all $j \in V_2$. This assumption is called the "non-modularity assumption". Then, a cluster can be represented by a graph that can be constructed as follows. The terms, "node", "vessel", and "rectangle", will be used interchangeably.

Step 0: Make a node for each vessel in the cluster. Let $S = \emptyset$, and let T be the set of all the rectangles in the cluster.
Step 1: From T, select such rectangles that have no rectangles on their right-hand sides. Include the selected rectangles into the set S.
Step 2: Select a rectangle from S whose left-hand side is located in the rightmost location among all rectangles in S. Let the selected rectangle be a. Identify rectangles in T whose right-hand sides are in contact with the left side of the rectangle a. Connect directed arcs from node $a$to the corresponding nodes of the identified rectangles.
Step 3: Eliminate rectangle a from sets S and T. Include rectangles corresponding to newly connected nodes into set S. If no rectangles remain in set S, then stop. Otherwise, go to step 2.

The graph for a y-cluster can be constructed in the same way as described above. Fig. 10 illustrates an x-cluster and the resulting cluster-graph.

Property A-1. *A cluster graph is a tree.*

Proof. By the non-modularity assumption, no two paths that begin from one node can merge into another node. Thus, the conclusion holds. □

Appendix B. Procedures for checking the stability of clusters

In order to confirm the stability of an x-cluster, it must first be confirmed that the movement of the x-cluster in either of right- and left-hand directions cannot reduce the total cost of rectangles

in the cluster. The following suggests a procedure for checking the direction of the cost change for the movement of an x-cluster.

B.1. A procedure to check the stability of an x-cluster against movement in the left-hand direction (Check-stab-x-left)

Step 0: (Initialize) Draw the cluster-graph and write weights with a plus sign on nodes whose lowest-cost points are the current positions of the nodes or are in the right-hand side of the current position. Write weights with a minus sign on nodes whose lowest-cost points are in the left-hand side of the current positions of the nodes. The minus sign implies that the movement of the cluster in the left-hand direction will reduce the cost of the corresponding rectangle. The plus sign implies that the cost of the corresponding rectangle increases by the movement of the cluster in the left-hand direction.

Step 1: (Roll-up tips of the cluster-graph)

Step 1.1: If the cluster-graph consists of a single node, go to step 2. Otherwise, select a tip from the cluster-graph. Go to step 1.2.

Step 1.2: If the selected tip is connected to the tail of an arc and has a positive weight, then, to make a new cluster, detach the selected tip and the nodes included in the selected tip from the rest of the cluster. If the selected tip is connected to the head of an arc and has a negative weight, then, to make a new cluster, detach the selected tip and the nodes included in the selected tip from the rest of the cluster. Go to the beginning of step 1.1.

Step 1.3: If conditions of step 1.2 do not hold, add the weight of the tip to the weight of the node next to the tip. In this case, we say that the selected tip is included in the node next to the tip on the cluster-graph (tree). From the graph, delete the tip and the arc connected to the tip. Go to step 1.1.

Step 2: (Judge the stability) If the weight of the last node is positive, then the x-cluster is stable against movement in the left-hand direction. If the weight is negative, then the x-cluster is said to be unstable against movement in the left-hand direction. That is, moving the cluster in the left-hand direction will reduce the cost of the cluster. If the weight of the last node is zero, the total cost will not change until a rectangle in the cluster arrives at its lowest-cost point during the movement of the cluster in the left-hand direction.

B.2. A procedure to check the stability of an x-cluster against movement in the right-hand direction (check-stab-x-right)

A procedure similar to the previously described procedure can be used to check the stability of an x-cluster against movement in the right-hand direction.

B.3. A procedure to check the stability of a y-cluster against movement in the downward direction (check-stab-y-down)

Suppose that the berthing time of a vessel in a cluster is greater than $d_i - b_i$. And suppose that no lower side of a rectangle in the cluster is in contact with the arrival time of the corresponding

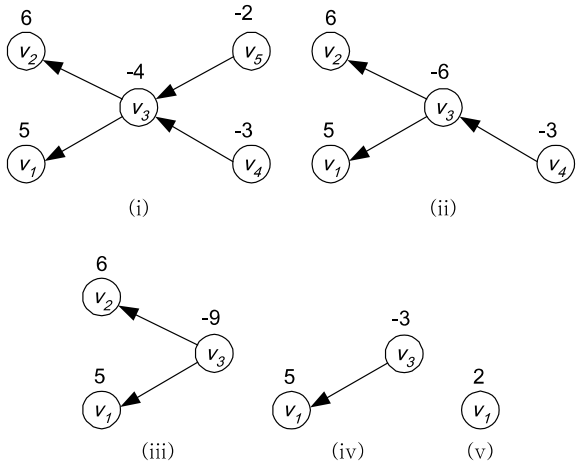


Fig. 11. An illustration of *check-stab-left*.

vessel or is zero. Then, according to Property 2, the total cost can be reduced by decreasing all the berthing times in the cluster until either of the two conditions in Property 2 is satisfied.

Fig. 11 illustrates the step-by-step process of *check-stab-x-left* for the cluster in Fig. 10. In (i), tip v_5 is selected. Because tip v_5 has a negative weight, its weight is added to that of v_3. Then, v_4 is selected whose weight is again added to that of v_3. Next, v_2 is selected, and, finally, v_1 is selected. Because the final node, v_1, has a positive weight, it can be concluded that the cluster is stable against movement in the left-hand direction.

Appendix C. Procedures to stabilizing clusters

This sub-section discusses how to stabilize clusters.

C.1. A procedure to stabilize an x-cluster against movement in the left-hand direction

The fact that a cluster is unstable implies that the total cost of the cluster can be reduced by moving the cluster toward either of the right- and left-hand directions. The problem of stabilizing x-clusters is similar to the problem of inserting idle times between jobs on a single machine for minimizing the total earliness and tardiness penalties (Fry et al., 1987). The difference is that the structure of a cluster graph of the single-machine scheduling problem is a chain, while the one in this paper is a tree. The following suggests a procedure for stabilizing a cluster.

Step 1: By using *check-stab-x-left*, check if the cluster is stable against movement in the left-hand direction. If stable, stop. Otherwise, go to step 2.

Step 2: Let s_0 be the minimum of the positive values among x-coordinates of the current locations deducted by the lowest-cost x-coordinate of the corresponding rectangle in the cluster. Find the longest distance possible that the cluster can move in the left-hand direction without making a rectangle in the cluster contact with the boundary of the berth or a rectangle

that is not a member of the cluster. Let the longest distance be s_1 in the first case and s_2 in the second case. Select the smallest value among s_0, s_1, and s_2. Move all the rectangles in the cluster to the left-hand direction by the distance of the selected value. If the selected element is s_1, stop. If the selected element is s_2, make a new cluster by additionally including the contacted rectangle into the current cluster. Go to step 1.

Note that the members of clusters may change after each iteration of step 2. The procedure to make an x-cluster stable against movement in the right-hand direction is similar to step 1.2, as described above.

C.2. A procedure to stabilizing a y-cluster against movement in the downward direction

If a cluster is unstable, then the total cost can be reduced by moving all the rectangles in the cluster downward until the berthing times of all the rectangles in the cluster is less than or equal to $d_i - b_i$, or a rectangle in the cluster is in contact with the arrival time of the corresponding vessel or in contact with the horizontal axis.

References

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

Cho, D.W., 1982. Development of a Methodology for Containership Load Planning. Ph.D. Dissertation, Oregon State University.

Cojeen, H.P., Dyke, P.V., 1976. The automatic planning and sequencing of containers for containership loading and unloading. In: Pitkin, Roche, Williams, (Eds.), Ship Operation Automation. North-Holland Publishing Co., Amsterdam.

Daganzo, C.F., 1989. The crane scheduling problem. Transportation Research Part B 23 (3), 159–175.

Francis, R.L., McGinnis Jr., R.F., White, J.A., 1992. In: Facility Layout and Location: An Analytic Approach. Prentice-Hall, Englewood Cliffs, NJ, pp. 189–199.

Fry, T.D., Armstrong, R.D., Blackstone, J.H., 1987. Minimizing weighted absolute deviation in single machine scheduling. IIE Transactions 19 (4), 445–450.

Gifford, L.A., 1981. Containership Load Planning Heuristic for a Transtainer-Based Container Port. M.Sc. Thesis, Oregon State University.

Heragu, S.S., Kusiak, A., 1991. Efficient models for the facility layout problem. European Journal of Operational Research 53, 1–13.

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

Kim, K.H., Kim, K.Y., 1999. An optimal routing algorithm for a transfer crane in port container terminals. Transportation Science 33 (1), 17–33.

Lai, K.K., Shih, K., 1992. A study of container berth allocation. Journal of Advanced Transportation 26 (1), 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 scheduling problem. Operations Research Letters 22, 105–110.

Perterkofsky, R.I., Daganzo, C.F., 1990. A branch and bound solution method for the crane scheduling problem. Transportation Research Part B 24 (3), 159–172.

Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A., Trevino, J., 1996. Facilities Planning, 344–348.