

Positioning empty containers among multiple ports with leasing and purchasing considerations

Il-Kyeong Moon · Anh-Dung Do Ngoc · Yun-Su Hur

Published online: 10 March 2010
© Springer-Verlag 2010

Abstract With the recent development of container transportation, the imbalance of empty containers among ports has become more serious. We consider the problem of positioning empty containers. The goal of this study is to propose a plan for transporting empty containers between container ports (terminals) to reduce the imbalance. There is currently a demand at each port and any backlog of containers is not permitted. The objective is to minimize the total relevant costs such as transportation cost, handling cost, and holding cost, etc. In this study, we develop a model with respect to the leasing and purchasing of containers. Mixed integer programming and genetic algorithms are used to solve the model. A hybrid GA is also proposed to reduce the computation time while still obtaining an acceptable result.

Keywords Empty container positioning · Hybrid genetic algorithm · Mixed integer programming

1 Introduction

The transportation of goods has become more developed with the acceleration of the world economy and the consequent increase in global trade. One of the most effective and least expensive commodities for transportation is containers. For inland transportation, there are several options for transporting goods, but for ocean transportation, only containers and container vessels can be used. A container, which is fully loaded with

I.-K. Moon (✉) · A.-D. Do Ngoc
Department of Industrial Engineering, Pusan National University, Busan 609-735, Korea
e-mail: ikmoon@pusan.ac.kr

Y.-S. Hur
Urban Creation Research Group, Busan Development Institute, Busan, Korea

goods from the supplier, is transported across oceans to the destination port and is then delivered to the customer and unloaded. After unloading, the container is maintained and stored at the destination port to await another trip. When there is an imbalance between the number of import containers and the number of export containers, some ports have a surplus of empty containers while other ports have a deficit. When ports have a deficit of empty containers, shipping companies must lease or purchase empty containers, and when ports have a surplus, the empty containers are stored in depots. Strategically, positioning the empty containers is, therefore, one of the most effective ways to solve the problem of the imbalance of containers. In 1995, the flow from Asia to the USA was 4 million 20-foot equivalent units (TEUs) and 3.5 million TEUs in the reverse direction. By 2005, the annual flow had increased to 12.4 million TEUs from Asia to the USA and 4.2 million TEUs in the reverse direction. By 2007, the annual flow was 15.4 million TEUs from Asia to the USA and 4.9 million TEUs in the reverse direction. (source: <http://people.hofstra.edu/geotrans/eng/ch3en/conc3en/worldcontainerflows.html>). Hence, the annual container flow from Asia to the USA increased by 8.4 million TEUs, in the 10 years from 1995 to 2005 and by 3 million TEUs in the 2 years from 2005 to 2007. In addition, the imbalance of container flow between Asia and the USA has also increased. The imbalance in 1995 was 0.5 million TEUs, that in 2005 was 8.2 million TEUs, and that in 2007 was 10.5 million TEUs. These numbers show that container flows and imbalance are continuously increasing. The situation is similar for container flows between Asia and Europe. Several approaches have been used by shipping companies to manage the imbalance of container flows and satisfy the demand. Among them, empty container positioning and leasing are widely considered. There have been many studies on empty container positioning, some of which focus on the leasing of containers, but they concentrate on long-term leasing only.

In this paper, we consider the problem of ocean positioning planning. In addition, to meet the demand, we also consider short-term leasing and purchasing factors. In long-term leasing, we can consider the leased container to be an owned one, assuming that the lease is for a considerably long period of time. However, in short term leasing, the lease duration is short (in several periods), and container handling comes into the picture when the container is being returned. The objective of this study is to minimize the total relevant costs, including transportation cost, handling cost, inventory holding cost, leasing cost, and purchasing cost. The remaining content of the paper is organized as follows: in Sect. 2, we develop the mathematical model. Two solution algorithms are presented in Sect. 3. Section 4 illustrates the computational results, and conclusion is presented in the final section.

2 The model

2.1 Literature review

Recently, many studies have been carried out concerning empty container positioning. Crainic et al. (1989) considered the multi-commodity location problem with balancing requirements. They proposed models for multi-commodity capacitated location

problems (MCLB) with an inter-depot balancing requirement. The decision variables of MCLB included a set of binary variables which decides on the opening or closing of depots and a set of continuous variables that presents the empty container flows among supply customers, depots, and demand customers. The objective function is to minimize the total cost, which involves the cost of opening the depot and transportation costs. The demand for empty containers must be satisfied. There have also been many studies carried out that attempt to solve the MCLB problem. Crainic and Delorme (1993) developed dual-ascent procedures for the proposed model. Crainic et al. (1993) solved the problem by using the Tabu search procedure. Gendron and Crainic (1997) presented the parallel branch and bound approach, which is based on a dual-ascent procedure previously proposed by Crainic and Delorme (1993). Gendron et al. (2003) also solved the problem using the Tabu search procedure, but they used the slope scaling method to find the starting solution. Li et al. (2004) studied the management of empty containers in a port with stochastic demand. Their analysis is based on the multistage inventory problem and the Markov decision processes with discrete time. They focused on the optimization of the pair-critical policy (U, D). In this policy, if the number of empty containers at a port is less than U , the empty containers are imported up to the amount of U ; and if it is more than D , the empty containers are exported down to the amount of D . Recently, Li et al. (2007) have since advanced the problem for the multi-port. Shen and Khoong (1995) proposed a decision support system (DSS) for empty container distribution planning. The DSS is based on the network optimization models. In the network, they considered the leasing-in, off-leasing, positioning-in, and positioning-out at a port. The problem is decomposed into three levels, namely the terminal (port) planning, intra-regional planning, and inter-regional planning. They considered the single type of containers.

2.2 Problem definition

First, the activity of a shipping company is described by considering the following scenario. At a port, the shipping company receives orders from customers. After receiving an order, the company transports empty containers to the customer. The customer's cargo will then be loaded into the containers. These full containers are then transported to the terminal and carried by a vessel to the destination port. At the destination port, the full containers are delivered, the cargo is unloaded, and the container is transported back to the depot. After being checked and maintained, these empty containers are ready for a new trip. When the full containers are transported to other ports, they are exported whereas if the full containers are received from another port, they are imported. If the number of imported containers is either less than or greater than that of exported containers, there will be an imbalance. It can be, therefore, seen that if the number of exported containers is greater than that of imported containers, there will be a shortage of empty containers. For the reverse situation, there would be a surplus of empty containers. At the surplus ports, the surplus containers will be stored in the depot and the cost for holding these containers must be paid. At the ports, where there is a shortage of containers, empty containers must be leased or purchased to

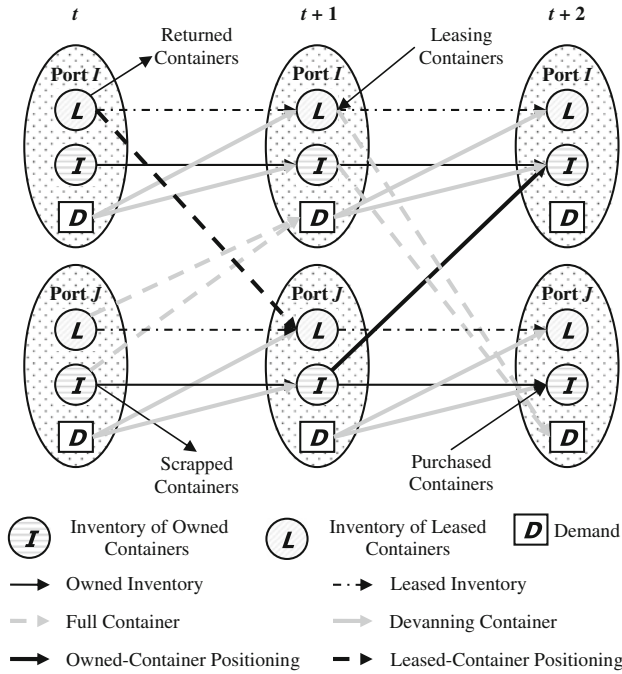


Fig. 1 The flow of containers between ports in multi periods

fulfill demand. The shipping company needs to find a way to reduce these costs. One solution involves empty containers being transported from ports with surplus containers to ports with a shortage. Transporting empty containers from port to port is termed “empty container positioning”.

When empty containers are positioned, the holding cost, leasing cost, and purchasing cost can be reduced, but there is a transportation cost. Moreover, because of the fluctuation in demand, a port may have a surplus at one time and a shortage another. Therefore, the shipping company must decide how many empty containers should be positioned, leased, and purchased at each port at various times. We therefore aim to propose a mathematical model to solve this problem.

Figure 1 graphically describes the situation. Continuous arrows represent the flow of empty containers, while dashed arrows represent the flow of full containers. At any port, the demand is fulfilled by empty containers in the inventory. There are two kinds of containers: owned containers and leased containers. Therefore, the inventory is divided into: the owned container inventory and leased container inventory. A number of continuous arrows connect the two types. In Fig. 1, node “ I ” denotes an owned-container inventory, while node “ L ” denotes a leased-container inventory. The arrows connecting the inventory of a port to itself ($I \rightarrow I$ and $L \rightarrow L$ at the same port) in the next period represent the flow of inventory to the next period. The arrows connecting the inventory of one port to that of another port ($I \rightarrow I$ and $L \rightarrow L$ for different ports) represent the flow of positioned empty containers.

The dashed arrows from the inventory of one port to the demand node of another port in the next period ($I \rightarrow D$ and $L \rightarrow D$) represent the flow of full containers, which are used to satisfy the demand. However, in the mathematical model, this does not mean that the full containers will arrive at the destination port after leaving the departure port in one period. They may arrive after several periods, depending on the transportation time. These flows represent the transport activity, and there are several kinds of costs for this activity. We divide the costs into fixed and variable costs. Fixed costs do not depend on the number of transported containers. The operation cost of a vessel is an example. This cost depends on the locations of the original and destination ports. The administration cost is also in this category. Variable costs depend on the number of transported containers. There is also a cost for loading and unloading containers at a certain port. This kind of cost is referred to as a handling cost and has both fixed and variable components.

As mentioned above, after the full containers have been delivered to customers, they are unloaded. The containers are then checked, maintained, and returned to the depot for the next trip. In this paper, these containers are referred to as “devanning containers” (the word “devanning” is used by Hyundai Merchant Marine Company). In Fig. 1, the arrows connecting the demand nodes to the inventory nodes ($D \rightarrow I$ and $D \rightarrow L$) at the same port in the next period represent the flow of devanning containers. We assume that the devanning time is one period. This means that full containers are emptied and returned to the inventory in the next period after they have arrived at the destination port.

There are a number of arrows that point away from or toward owned-container inventory nodes. The arrows pointing to the owned-container inventory in a certain period represent the flow of purchased containers. These purchased containers are bought from suppliers and they are first used in that period. The cost relating to this flow is the purchasing cost, and it depends on the container type. The arrows that point away from the owned-container inventory in a period represent the flow of scrapped containers. Scrapped containers are those that can no longer be used. These containers may have expired or be used for another purpose. There are also a number of arrows that point away from or toward leased-container inventory nodes. The arrows that point away represent the flow of returned containers. In leasing a container, the shipping company has to pay for activities that do not depend on the number of leased containers, such as administration. Hence, the leasing cost should include both fixed and variable costs. After the containers have been leased from the leasing company and used for one or more trips, they are returned to their owners. These containers are termed “returned containers”. The arrows that point toward leased-container inventory nodes represent containers leased in that period.

In the real world, there are several kinds of container types such as 20, 40 ft, dry, open-top or refrigerated containers. Thus, in this study, we consider multiple types of containers. Moreover, the short-term leasing of containers is considered, which means that containers can be leased and then returned to another port determined by the lessor. This short-term leasing consideration has not been considered in previous studies.

2.3 The mathematical model

A number of assumptions have been made as follows:

- The demand must be satisfied, and no backlog is permitted.
- The total capacity of owned vessels and rented vessels is enough to transport the full containers for a certain period.
- The route of the vessels is not considered.
- The full containers will be emptied after they have arrived at the destination port within one period.
- There is no limitation to the number of purchased containers within a period.

We propose a mathematical model as follows:

Indices:

- t : time period, $t = 1, 2, 3, \dots, T$
 i, j, k : port, $i, j, k = 1, 2, \dots, P$
 v : type of containers, $v = 1, 2, \dots, V$

Parameters:

Inventory

- K_{iv} : maximum inventory level of type v containers at port i
 H_{iv} : unit cost of holding type v containers at port i .

Leasing

- LK_{iv} : capacity of leasing type v containers at port i
 LF_{iv} : fixed leasing cost of type v containers at port i
 LV_{iv} : variable leasing cost of type v containers at port i
 RK_{iv}^k : maximum number of type v containers, which are leased at port k and that will be returned to port i

Transportation

- TF_{ij}^1 : fixed cost for transporting containers from port i to port j by owned vessels
 TV_{ijv}^1 : variable transportation cost of type v container from port i to port j by owned vessels
 TF_{ij}^2 : fixed cost for transporting containers from port i to port j by rented vessels
 TV_{ijv}^2 : variable transportation cost of type v container from port i to port j by rented vessels
 TK_{ijt}^1 : capacity of transportation from port i to port j by owned vessels in period t
 TK_{ijt}^2 : capacity of transportation from port i to port j by rented vessels in period t
 TI_{ij} : transportation time from port i to port j .

Handling

- RF_{ji}^1 : fixed handling cost of containers at port i , which are transported from port j by owned vessels

RV_{jiv}^1 : variable handling cost of type v containers at port i , which are transported from port j by owned vessels

RF_{ji}^2 : fixed handling cost of containers at port i , which are transported from port j by rented vessels

RV_{jiv}^2 : variable handling cost of type v containers at port i , which are transported from port j by rented vessels

Purchasing

PC_{iv} : unit cost of purchasing type v containers at port i .

Others

D_{ijvt} : demand for type v containers at period t , which should be transported from port i to port j

SC_{ivt} : number of scrapped containers at port i at period t , which are type v

TEU_v : changing rate from container type v to TEU.

Decision variables:

Inventory

I_{ivt} : inventory level of type v owned containers at port i at the end of period t

LI_{ivt}^k : inventory level of type v containers at port i in period t , which are leased at port k

Transportation

x_{ijvt}^e : number of type v empty containers (owned containers) leaving in period t for being transported from port i to port j

x_{ijvt}^f : number of type v full containers (owned containers) leaving in period t for being transported from port i to port j

x_{ijvt}^{elk} : number of type v empty containers (leased at port k) leaving in period t for being transported from port i to port j

x_{ijvt}^{flk} : number of type v full containers (leased at port k) leaving in period t for being transported from port i to port j

f_{ijvt}^1 : number of type v containers, leaving in period t for being transported from port i to port j by owned vessels

f_{ijvt}^2 : number of type v containers, leaving in period t for being transported from port i to port j by rented vessels

$$s_{ijt}^1 : \begin{cases} 1, & \text{if there is one or more vessels leaving in period } t \text{ for transporting} \\ & \text{containers from port } i \text{ to port } j \\ 0, & \text{otherwise} \end{cases}$$

$$s_{ijt}^2 : \begin{cases} 1, & \text{if there is one or more rented vessels leaving in period } t \text{ for} \\ & \text{transporting containers from port } i \text{ to port } j \\ 0, & \text{otherwise} \end{cases}$$

Leasing

- y_{ivt} : number of type v containers, which are leased at port i in period t
- w_{ivt}^k : number of type v containers, which are leased at port k , and will be returned to port i in period t
- u_{ivt} : $\begin{cases} 1, & \text{if we lease type } v \text{ containers at port } i \text{ in period } t \\ 0, & \text{otherwise} \end{cases}$

Purchasing

p_{ivt} : number of type v containers purchased at port i in period t .

The mathematical model:

$$\begin{aligned} \text{Min } Z = & \sum_{i=1}^P \sum_{j=1, j \neq i}^P \sum_{t > TI_{ji}}^T \left[\left(\text{RF}_{ji}^1 s_{ji,t-TI_{ji}}^1 + \sum_v \text{RV}_{jiv}^1 f_{jiv,t-TI_{ji}}^1 \right) \right. \\ & \left. + \left(\text{RF}_{ji}^2 s_{ji,t-TI_{ji}}^2 + \sum_v \text{RV}_{jiv}^2 f_{jiv,t-TI_{ji}}^2 \right) \right] \\ & + \sum_{i=1}^P \sum_{j=1, j \neq i}^P \sum_{t=1}^T \left[\left(\text{TF}_{ij}^1 s_{ij,t}^1 + \sum_v \text{TV}_{ijv}^1 f_{ijvt}^1 \right) \right. \\ & \left. + \left(\text{TF}_{ij}^2 s_{ij,t}^2 + \sum_v \text{TV}_{ijv}^2 f_{ijvt}^2 \right) \right] \\ & + \sum_{i=1}^I \sum_{v=1}^V \sum_{t=1}^T \left[H_{iv} \left(I_{ivt} + \sum_k \text{LI}_{ivt}^k \right) \right] \\ & + \sum_{i=1}^I \sum_{v=1}^V \sum_{t=1}^T (\text{LF}_{iv} u_{ivt}) + \sum_{k=1}^P \sum_{i=1}^P \sum_{v=1}^V \sum_{t=1}^T \text{LV}_{kv} \text{LI}_{ivt}^k \\ & + \sum_{k=1}^P \sum_{i=1}^P \sum_{j=1, j \neq i}^P \sum_{v=1}^V \sum_{t=1}^T \left[\text{LV}_{kv} \left(x_{ijvt}^{elk} + x_{ijvt}^{flk} \right) TI_{ij} \right] \\ & + \sum_{i=1}^I \sum_{v=1}^V \sum_{t=1}^T \text{PC}_{iv} p_{ivt} \end{aligned}$$

Subject to

$$\begin{aligned} I_{ivt} = & I_{iv,t-1} + \sum_{\substack{j=1, \\ j \neq i}}^P \sum_{\substack{t > TI_{ji} \\ t > TI_{ji+1}}}^T x_{jiv,t-TI_{ji}}^e + \sum_{\substack{j=1, \\ j \neq i}}^P \sum_{\substack{t > TI_{ji+1} \\ t > TI_{ji}}}^T x_{jiv,t-TI_{ji}-1}^f \\ & - \sum_{\substack{j=1, \\ j \neq i}}^P \left(x_{ijvt}^e + x_{ijvt}^f \right) + p_{ivt} - \text{SC}_{ivt} \quad \forall i, v, t \end{aligned} \tag{1}$$

$$\begin{aligned}
 LI_{ivt}^k &= LI_{iv,t-1}^k + \left(\sum_{\substack{j=1, \\ j \neq i}}^P \sum_{t > TI_{ji}}^T x_{jiv,t-TI_{ji}}^{elk} + \sum_{\substack{j=1, \\ j \neq i}}^P \sum_{t > TI_{ji}+1}^T x_{jiv,t-TI_{ji}-1}^{flk} \right) \\
 &\quad - \sum_{\substack{j=1, \\ j \neq i}}^P (x_{ijvt}^{elk} + x_{ijvt}^{flk}) - w_{ivt}^k \quad \forall i, v, t, k \neq i
 \end{aligned} \tag{2a}$$

$$\begin{aligned}
 LI_{ivt}^i &= LI_{iv,t-1}^i + \sum_{\substack{j=1, \\ j \neq i}}^P \sum_{t > TI_{ji}}^T x_{jiv,t-TI_{ji}}^{eli} + \sum_{\substack{j=1, \\ j \neq i}}^P \sum_{t > TI_{ji}+1}^T x_{jiv,t-TI_{ji}-1}^{fli} \\
 &\quad - \sum_{\substack{j=1, \\ j \neq i}}^P (x_{ijvt}^{eli} + x_{ijvt}^{fli}) + y_{ivt} - w_{ivt}^i \quad \forall i, v, t
 \end{aligned} \tag{2b}$$

$$x_{ijvt}^f + \sum_{k=1}^P x_{ijvt}^{flk} = D_{ijvt} \quad \forall i, j \neq i, v, t \tag{3}$$

$$I_{ivt} + \sum_{k=1}^P LI_{ivt}^k \leq K_{iv} \quad \forall i, v, t \tag{4}$$

$$y_{ivt} \leq LK_{iv} \cdot u_{ivt} \quad \forall i, v, t \tag{5}$$

$$w_{ivt}^k \leq RK_{iv}^k \quad \forall i, v, t, k4 \tag{6}$$

$$f_{ijvt}^1 + f_{ijvt}^2 = x_{ijvt}^e + x_{ijvt}^f + \sum_{k=1}^P (x_{ijvt}^{elk} + x_{ijvt}^{flk}) \quad \forall i, j \neq i, v, t \tag{7}$$

$$\begin{aligned}
 \sum_{v=1}^V (f_{ijvt}^1 \cdot TEU_v) &\leq TK_{ijt}^1 \cdot S_{ijt}^1 \quad \text{and} \quad \sum_{v=1}^V (f_{ijvt}^2 \cdot TEU_v) \\
 &\leq TK_{ijt}^2 \cdot s_{ijt}^2 \quad \forall i, j \neq i, t
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 x_{ijvt}^e, x_{ijvt}^f, x_{ijvt}^{elk}, x_{ijvt}^{flk}, y_{ivt}, w_{ivt}^k, I_{ivt}, LI_{ivt}^k, f_{ijvt}^1, f_{ijvt}^2, p_{ivt} \geq 0, \text{ and } s_{ijt}^1, s_{ijt}^2, \\
 u_{ivt} = \{0, 1\}
 \end{aligned} \tag{9}$$

The objective function is to minimize the sum of the handling cost, transportation cost, holding cost, leasing cost, and purchasing cost. Constraints (1), (2a), and (2b) are inventory balance constraints. The following applies to the owned-container inventory:

- The number of empty containers entering the inventory is the sum of the number of empty containers in the previous period (period $t - 1$), the number of empty containers received in the current period, the number of full containers received in the previous period, and the number of containers purchased in the current period

- The number of empty containers leaving the inventory is the sum of the number of empty and full containers transported to other ports and the number of scrapped containers

The situation is similar for the leased inventory. The number of empty leased containers entering the inventory is the sum of the number of empty leased containers in the previous period (period $t - 1$), the number of empty leased containers received in the current period, the number of full leased containers received in the previous period, and the number of newly leased containers. The number of empty leased containers leaving the inventory is the sum of the number of empty and full leased containers transported to other ports and the number of returned containers that were leased in previous periods.

Constraint (3) ensures that demands are always satisfied. Constraint (4) avoids the inventory level that exceeds the maximum inventory level. Constraints (5) and (6) limit the number of newly leased containers and the number of returned containers, respectively. Constraints (7) and (8) relate to the vessel. Constraint (7) balances the total number of transported containers. In the expression of this constraint, the left-hand side represents the total number of transported containers, which is expressed in terms of the vessel, while the right-hand side represents the total number of owned and leased containers. Constraint (8) ensures that the number of transported containers does not exceed the vessel capacity. The vessel capacity is defined in terms of TEU while the weight capacity is not considered. As in many papers on empty container movement, especially those written by Crainic et al. (1989, 1993) and Crainic and Delorme (1993), the variables that relate to the flow of empty containers are considered as continuous variables in this paper. Moreover, the values of these variables are large and can be rounded to the nearest integer.

3 LP-based genetic algorithm and hybrid genetic algorithm

The above mathematical model is a Mixed Integer Programming (MIP) model. However, the described problem is conjectured as an NP-hard problem and it has been shown that it is time-consuming to solve this type of a flow model. We therefore develop two meta-heuristic algorithms.

Drawing from the work of Nemhauser and Wolsey (1988), we can present the MIP problem in a general form as follows:

$$\begin{aligned}
 \text{(MIP) Min} \quad & cx + hy \\
 \text{Subject to} \quad & Ax + By \geq D \\
 & x \in \{0, 1\}, y \geq 0
 \end{aligned}$$

If the value of $x = \bar{x}$ can be fixed, the problem becomes

$$\begin{aligned}
 \text{(LP}(\bar{x})\text{) Min} \quad & hy + C && \text{where } C = c\bar{x} \\
 \text{Subject to} \quad & By \geq D - A\bar{x} \\
 & y \geq 0
 \end{aligned}$$

The $LP(\bar{x})$ is a linear programming problem, which can be easily solved by using a range of optimization software. If $LP(\bar{x})$ is feasible and its solution is \bar{y} , then (\bar{x}, \bar{y}) will be a feasible solution of the MIP problem. If Z_{MIP} is the optimal value of the MIP problem, and $Z_{LP}(\bar{x})$ is the optimal value of the $LP(\bar{x})$ problem, then $Z_{LP}(\bar{x}) \geq Z_{MIP}$. When a meta-heuristic is used to generate the value of integer variables, then solving the LP problem gives the value of continuous variables. The optimum solution of the MIP problem is denoted as (x^*, y^*) . If a value \bar{x}_k exists so that $\bar{x}_k = x^*$ in the generated set, then the algorithm can provide us with an optimum solution for the MIP problem. However, the optimum solution cannot always be found using a meta-heuristic method; in most cases, this method can only provide the best solution among the available feasible solutions. In this paper, a genetic algorithm, which is a well-known meta-heuristic method, is used to determine the values of integer variables such that the corresponding solution comprising a set of continuous and integer variables is near optimal.

There have been several papers on the combination of a GA with Linear Programming (LP). [Lou et al. \(2001\)](#) proposed an integrated method to solve the general MIP problem and applied it to production planning and scheduling in a batch (PPS) problem. They integrated ordinal optimization and LP together with a GA. In this approach, the chromosome represents the solution of integers and continuous variables. When a chromosome is created, its feasibility is checked. If the solution according to a certain chromosome is infeasible, that chromosome must be re-created. This is a disadvantage since it may take time to re-create chromosomes. Earlier, [Rajasekharan et al. \(1998\)](#) also considered combining a GA with MIP to solve the facility layout problem (FLP) in a flexible manufacturing system (FMS). They suggested using a GA to find the values of some integer variables, and then to find the values of the remaining integer variables and continuous variables by applying an LP-based branch and bound procedure. The disadvantage of this method is that in the first phase, a chromosome cannot give the values for all integer variables. Therefore, it may take a long time in the second phase to execute the LP-based branch and bound procedure, instead of implementing the LP.

[Pedroso \(1998\)](#) proposed a method for solving this MIP problem, in which the values of integer variables are determined by an evolutionary algorithm and the values of continuous variables are then determined by solving the LP problem. He also pointed out a way to address the infeasibility of the LP problem. In this paper, we propose two approaches to combine a GA with an LP. In the first approach, we apply the algorithm proposed by [Pedroso \(1998\)](#), while making a small change to the manner in which the infeasibility of the LP problem is handled. In Pedroso's algorithm, the chromosome that makes the relevant LP problem infeasible (which we refer to as an infeasible chromosome, as opposed to a feasible chromosome) is accepted and measured by considering its proximity to the feasible region. We, however, have chosen to apply a procedure called the LP-based GA, which adjusts the infeasible chromosome so that it becomes feasible. Our second approach-hybrid GA uses a heuristic approach to find a feasible solution, rather than the optimal solution of the LP problem. The algorithm that we propose can therefore be described as follows:

1. Generate a set of values of x : $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ by a meta-heuristic method.
2. Check the feasibility of the $LP(\bar{x}_i)$ problem. If the $LP(\bar{x}_i)$ problem is feasible, it is solved (heuristic procedure is executed in the second approach) to obtain

- \bar{y}_i and (\bar{x}_i, \bar{y}_i) is a feasible solution of the MIP problem. Otherwise, we modify \bar{x}_i so that it makes the LP problem feasible.
3. We then select the feasible solution that provides the smallest objective function value of the MIP problem. We consider this solution to be the best feasible solution found by the algorithm.

3.1 LP-based genetic algorithm

Here, we do not explain the theory of GAs. It is more relevant to this study to discuss how GAs are used to solve our model. The following section presents the structure of chromosome, fitness function, and the GA’s operators and how the experiment is conducted.

a. Encoding

In the mathematical model, there are three sets of binary variables, namely, s_{ijt}^1 , s_{ijt}^2 , and u_{ivt} . Decoding each chromosome gives us the values of these variables. The chromosome is coded as follows:

- The chromosome is a string that has three substrings.
- Each substring represents the value of the corresponding variables in a set. The first substring represents the value of s_{ijt}^1 , the second substring represents the value of s_{ijt}^2 and the third substring is used for u_{ivt} .
- Each component (or each gene) of a substring is symbolized by a variable. We can illustrate how a set of variables is mapped to a substring by using the substring that is represented by the set of variables s_{ijt}^1 as an example. From left to right along the substring, the first gene is represented by s_{111}^1 , the second gene is represented by s_{112}^1 , the T th gene is represented by s_{11T}^1 , the $(T + 1)$ th gene is represented by s_{121}^1 , and the last gene is represented by s_{PP}^1 . Therefore, the length of each substring determines the number of variables in its corresponding set. In detail, the formula to compute the chromosome length with respect to P , V , and T is $2 * P^2T + PVT$. The value of each gene is 0 or 1.

The following example is the string for the case where the number of ports is 2, the number of container types is 2, and the number of period is 4.

From the chromosome presented in Fig. 2, we can see that $s_{111}^1=0$, $s_{112}^1=1$, $s_{223}^1=1$, $s_{112}^2=0$, $s_{224}^2=1$, $u_{111}=1$ and $u_{223}=0$. In order to create the initial population, we assigned the random value (0 or 1) to each gene.

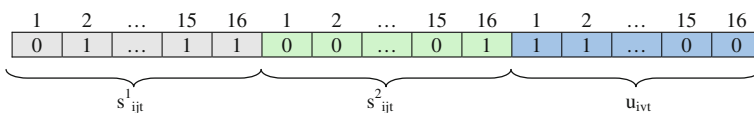


Fig. 2 Chromosome coding

b. Fitness function

In the maximization problem, the objective function value can be directly used for the fitness. However, our mathematical model is a minimization model. Therefore, if the objective function value of a chromosome is higher, that chromosome shows less fitness. In this research, the following formula is used to calculate the fitness.

$$\text{Fitness}(i) = \frac{\text{Max}Z - Z(i)}{\sum_{k=1}^N [\text{Max}Z - Z(k)]} \quad \text{Max}Z = \max_k = 1, \dots, N(Z(k)), \quad N: \text{population size}$$

where $Z(i)$ is the objective function value of chromosome i .

The denominator of the above formula will be zero when all chromosomes are identical. In this case, the GA is converged and the solution will be stopped. However, this case is very unlikely.

c. The solution algorithm

In this section, we discuss how a GA is applied to solve the problem.

Step 1. Generate the initial population.

In this step, a number of chromosomes are created for the first generation. As mentioned above, a value is randomly assigned to each gene of each chromosome. Subsequently, the feasibility of the solution must be checked. If the solution given by the chromosome is infeasible, the values of some of the genes must be changed. The infeasibility appears when demand is higher than the total capacity of the owned and rented vessels. We have two situations as follows:

- Case 1: $s_{ijt}^1 = 1, s_{ijt}^2 = 0 (i \neq j)$, and demand is higher than the capacity of the owned vessel; we set $s_{ijt}^2 = 1$.
- Case 2: $s_{ijt}^1 = 0, s_{ijt}^2 = 0 (i \neq j)$, and there is a demand from port i to port j in period t ; we set $s_{ijt}^1 = 1$ and continue to check as the first case.

Step 2. Calculate the objective function value and fitness.

In this step, each chromosome represents the value of the binary variables. Therefore, we will solve the LP(\bar{x}) for each chromosome. The feasibility of the LP(\bar{x}) is guaranteed because we assume that there is enough vessel capacity for the demand and there is no limit in the number of purchased containers. In this research, LINGO is used to solve this LP problem. After solving the problem, a solution to the continuous variables can be found and the objective function value and the fitness of each chromosome can easily be calculated.

Step 3. Update the solution and check for the stopping criteria.

In the population, the current best chromosome is chosen and its objective function value is compared with that of the best chromosome that has been found so far. For easy understanding, the term “the current best chromosome” is used to refer to the chromosome that gives the smallest objective function among a certain population, and the term “the best chromosome” is used to denote the chromosome that gives the best solution found since the beginning of the GA execution. If the best chromosome has an objective function value

greater than that of the current best chromosome, the best chromosome is updated.

Subsequently, the stopping criterion should be checked. The criterion might either demand that a maximum number of generations has been reached or that there is no improvement after a number of consecutive generations. In this research, we use the first criterion.

If the stopping criterion is satisfied, the best chromosome provides the solution. Otherwise, we go to step 4.

Step 4. Create a new generation.

In this step, the GA operators are performed to create a new generation. The procedure is outlined in the following steps:

- Cloning
- Selection and crossover

In this research, the roulette wheel and tournament are used for the experiment on GA parameters. After the experiment, the tournament is selected. For crossover, since the string of each chromosome has three substrings, a random crossover operator will be performed on each of them. We based our decision to use random crossover operators on the results of experiments on GA parameters. Each of the child's genes is randomly inherited from one of two parent chromosomes.
- Mutation

In this research, a uniform method is used in which the value of a gene is changed from 0 to 1 or from 1 to 0 if it is selected for mutation.
- Checking the feasibility of the new chromosome

The population size of the new generation is the same as that of the old generation.
Return to step 2.

3.2 Hybrid genetic algorithm

As mentioned above, the LP-based GA is able to solve larger problems than the MIP can. However, it must solve an LP problem for each chromosome. While solving a single LP problem is not time consuming, solving an extended series of LP problems is. The LP problem needs to be solved for each chromosome because the chromosome fitness needs to be more precisely evaluated. Unfortunately, it is seen that this process takes too much time in the computational experiment section. Therefore, we propose using a heuristic method rather than an optimization software to solve the LP problem. Because the heuristic method can find a feasible solution of the LP problem instead of the optimal solution, it reaches a conclusion more rapidly though less precisely than the LP-based method. The procedure for the heuristic method is as follows:

a. Heuristic algorithm for finding a feasible solution for the LP problem

Step 1. Assign the empty containers to fulfill the demand.

In this step, we try to fulfill the demand based on the number of on-hand empty containers. The procedure of this step is as follows:

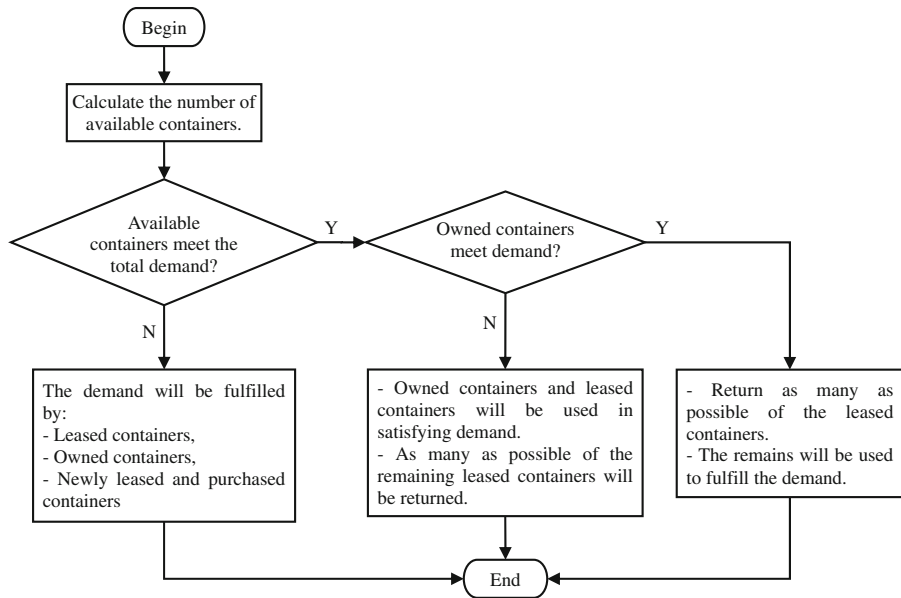


Fig. 3 Flowchart of step 1

- Step 1.1. Calculate the number of available containers.
- Step 1.2. If owned empty containers can satisfy the total demand,
 - Return as many as possible of the leased containers.
 - Assign remaining containers to fulfill the demand. The leased containers have a higher priority for assigning. An attempt is made to assign the leased containers to the destination to which they can be returned.
 - Go to step 1.5
 Otherwise, go to the next step.
- Step 1.3. If available containers can satisfy the total demand, then
 - Owned containers and leased containers are used to fulfill demand. As for the previous step, an attempt is now made to assign the leased containers to the destination which they can be returned.
 - As many as possible of the remaining leased containers will be returned.
 - Go to step 1.5.
 Otherwise, go to the next step.
- Step 1.4. All the available containers will be assigned to fulfill part of the demand. The remaining demand will be satisfied by newly leased and purchased containers. As for the above step, an attempt is made to assign the leased containers to the destination to which they can be returned.
- Step 1.5. End of step 1.

The flowchart for this step is shown in Fig. 3

Step 2. In this step, all ports are checked to ensure that they have all been considered. If all ports have been considered, we go to the next step. Otherwise, the next port is considered and we return to step 1.

Step 3. Position empty containers.

After fulfilling the demand at all ports, it can be easily determined which ports have a surplus of empty containers. With these ports, the empty containers will be positioned to other ports with random amounts.

The procedure is as follows:

- Step 3.1. Calculate the total demand and estimate the number of available containers in the next period. The available containers in the next period include the surplus empty containers in the current period, the empty containers that will arrive in the next period, and the full containers that will arrive in this period.
- Step 3.2. If the number of available containers is not greater than the total demand, we proceed directly to step 3.3. Otherwise, we first calculate the number of containers that are available for positioning by subtracting the total demand from the number of containers available in the next period. The remaining containers will be assigned for positioning to other ports with random amounts.
- Step 3.3. If all ports have been considered, we stop and go to step 4. Otherwise, the next port is chosen and we return to step 3.1.

The flowchart of step 3 is shown in Fig. 4

Step 4. If all periods have been considered, we stop. Otherwise, we return to step 1.

b. The solution algorithm

The above heuristic algorithm can be applied to approximate the optimal solution of the LP problem for each chromosome and the GA can be executed to solve the IP problem. It is noted that the solution obtained by the heuristic algorithm is a feasible one, rather than an optimal one. We therefore need to moderately alter the solution algorithm. In the LP-based GA approach, there is only one current best chromosome. However, we do not estimate exactly the fitness of the chromosome in the hybrid-GA. Hence, instead of using the best chromosome, a set of good chromosomes is considered and the best chromosome among them is chosen after we solve the LP problem for each of them using optimization software (Lou et al. 2001). The set of good chromosomes includes some chromosomes that have a smaller objective function value than other chromosomes generated since the beginning of the GA execution. Therefore, in each generation, after calculating the fitness of each chromosome, the set of good chromosomes is updated. Note that the size of this set does not change. Thus, if one or more chromosomes in the current population are better than the chromosomes in the set of good solutions, they are added to the set and the worst chromosomes in the set are omitted.

After the GA execution is complete, the set of good chromosomes is known, and LINGO software is used to solve the LP problem for each chromosome. The chromosome that gives the best objective function value is selected and the best solution is derived from this chromosome.

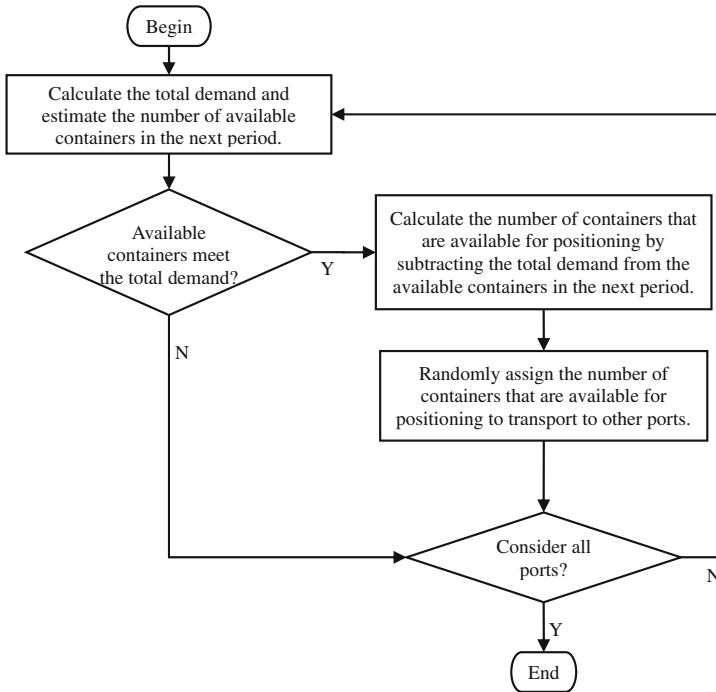


Fig. 4 Flowchart of step 3

The solution algorithm of the hybrid GA is similar to that of the LP-based GA and is presented as a flowchart in Fig. 5. First, chromosomes are generated for the initial population. The fitness of each chromosome is then calculated. Objective function values that are derived from chromosomes are used to compute the fitness of each chromosome. In this step, rather than using LINGO to solve the LP problem, we use a heuristic algorithm, which has been mentioned above. The next step is updating the set of good chromosomes. In this step, rather than updating the best chromosome, we update the set of good chromosomes. The objective function value of each chromosome (in the population of the current generation) is compared with the values of chromosomes in the set of good chromosomes. Better chromosomes are added to the set of good chromosomes, while the worst ones are removed. Afterward, the stopping criterion is checked. If the stopping criterion is not satisfied, a new generation is created by GA operators such as crossover and mutation. Otherwise, GA execution is stopped and the next step is to find the best chromosome among those in the set of good chromosomes. The chromosome that gives the best objective function value is selected. In this step, LINGO is used to solve the LP problem in order to find the objective function value.

4 Computational results

Initial experiments are conducted to show the performance of the two proposed algorithms. Three problems were generated with a range of sizes, as presented in Table 1.

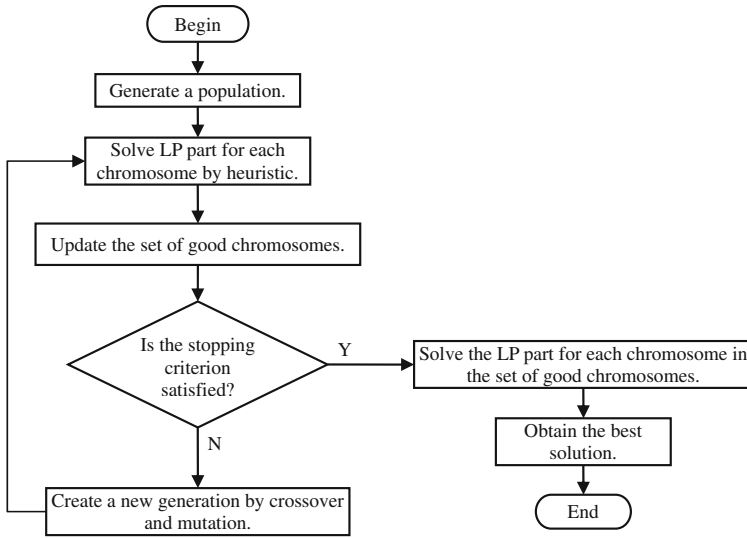


Fig. 5 Flowchart of the hybrid GA

Table 1 Problem size for initial experiments

Problem instance	Problem size			Number of variables			Number of Constraints
	Number of ports	Number of container types	Number of periods	Integer variables	Continuous variables	Total variables	
I-1	3	4	8	240	3,744	3,984	1,344
I-2	3	4	24	720	11,232	11,952	4,032
I-3	5	4	30	2,100	49,800	51,900	13,800

All the experiments were conducted on a computer having an Intel Pentium 4 3.4 GHz processor and 1 GB RAM. LINGO was used to solve the LP part. Table 2 shows the comparison between all three algorithms. The computation times are given in seconds. A comparison of the objective values is provided in the last column of the table, and the differences in objective values are expressed as percentages. These initial experiments on the computation times and the problem size have shown that the results obtained by the LP-based GA and by MIP are approximately the same for problems of small size, and that the LP-based GA can solve the problem with the largest size, while MIP cannot. However, the computation time is too long. Note that this study targets the operations of merchant marine companies, and taking decisions on the repositioning of empty containers is a part of their day-to-day operations. Therefore, we restrict the acceptable time to be less than a day. If the computation time is longer than a day, we can safely say that it is too long. LINGO is used for each chromosome to solve the LP problem. Therefore, in each run, LINGO is used 40×200 , i.e., 8,000 times (40 is the population size and 200 is the maximum generations). Even though the time for solving each problem is short, the number of times LINGO is used is excessive. For

Table 2 Comparison between MIP, the LP-based GA, and the hybrid GA

Problem instance	MIP	LP-based GA (s)	Hybrid GA (s)	Solution (% penalty)
I-1	<1 min	1,801	84	0.81% (vs. MIP) 0.81% (vs. LP-based)
I-2	3,105 s	9,807	228	0.75% (vs. MIP) 0.75% (vs. LP-based)
I-3	>3 days, cannot obtain result	52,051	1,056	0.56% (vs. LP-based)

this reason, use of the LP-based GA is too time consuming. In Table 2, a considerable improvement in the computation time is seen for the hybrid GA. The computation time is reduced by more than 20 times for the smallest problem and about 50 times for the largest. The objective function values of the hybrid GA are greater than those of the LP-based GA, but the differences are no more than 1%. This is an acceptable error. The hybrid GA, therefore, demonstrates efficiency in solving the larger problems.

We also randomly generated 20 problems that had different relevant costs, maximum inventory level, leasing capacity, demand, and other parameters. The number of ports and the number of container types are randomly generated in the ranges of [3, 10] and [4, 8], respectively. The numbers of periods in weeks are 13, 26, 39, and 52 (corresponding to the number of quarters in a year). The problem sizes are shown in Table 3. For each problem, the LP-based GA and the hybrid GA are run five times. The comparison among MIP, the LP-based GA, and the hybrid GA are presented in Table 4.

In Table 4, the cells containing a “—” symbol indicate that the corresponding problems cannot be solved within a day and results cannot be obtained. The computation times for the LP-based GA and the hybrid GA are the average computation times for five runs. The percentage penalty for the hybrid GA is calculated using the average objective value of five runs. It is seen that the hybrid GA is the superior method in terms of computation time for large problems. Moreover, when we compare the objective values, the percentage difference between the hybrid GA result and the results of the other methods (MIP and the LP-based GA) does not exceed 2%. Our results present five distinct problem instances that can be compared. Hence, to make the evaluation more convincing, we extend it by looking at the lower bound of each problem instance. In this paper, we use the objective function value of the linear relaxation problem as the lower bound of the original problem. The linear relaxation problem is derived from the original problem by converting all binary variables to continuous variables which changes the constraint $s_{ijt}^1, s_{ijt}^2, u_{ivt} \in \{0, 1\}$ to $0 \leq s_{ijt}^1, s_{ijt}^2, u_{ivt} \leq 1$. Table 4 also shows that the difference between the hybrid GA and the lower bound does not exceed 5% in all problem instances. This provides persuasive evidence that the hybrid GA performs effectively.

When the size of the problem increases, especially the number of ports, the computation time of the hybrid GA becomes too long. This can be explained as follows. There are two phases in the hybrid GA. The first is to find the good set of chromosomes

Table 3 Problem size for 20 randomly generated problems

Problem instance	Problem size			Number of variables			Number of constraints
	Number of ports	Number of container types	Number of periods	Integer variables	Continuous variables	Total variables	
1	6	8	13	1,560	69,264	70,824	16,380
2	10	5	13	3,250	170,950	174,200	28,990
3	5	6	13	1,040	32,370	33,410	8,710
4	4	8	13	832	24,544	25,376	7,384
5	9	7	13	2,925	179,361	182,286	32,175
6	8	6	26	4,576	223,392	227,968	44,096
7	9	5	26	5,382	256,230	261,612	47,034
8	4	6	26	1,456	36,816	38,272	11,232
9	6	5	26	2,652	86,580	89,232	21,060
10	6	4	26	2,496	69,264	71,760	17,160
11	5	8	39	3,510	129,480	132,990	34,320
12	3	5	39	1,287	22,815	24,102	8,073
13	5	7	39	3,315	113,295	116,610	30,225
14	6	6	39	4,212	155,844	160,056	37,440
15	7	7	39	5,733	273,273	279,006	58,695
16	3	7	52	2,028	42,588	44,616	14,820
17	4	5	52	2,704	61,360	64,064	18,928
18	8	4	52	8,320	297,856	306,176	60,736
19	4	7	52	3,120	85,904	89,024	26,000
20	6	6	52	5,616	207,792	213,408	49,920

and it is performed by GA operations, while the second is to solve the LP part of each chromosome of this good set by using optimization software. The computation time in the first phase is short but that in the second phase is too long. Since the time to solve an LP problem has a polynomial relationship with the size of the problem, a large amount of time is required when the size of the problem is large. For example, in problem instance 6, the time to solve the LP part (i.e. the second phase of the hybrid GA) accounts for more than 70% of the total computation time, whereas it accounts for approximately 81% in problem instance 2 and 88% in problem instance 7. Moreover, the number of continuous variables is directly proportional to the cube of the number of ports, and the number of constraints is directly proportional to the square of the number of ports. Therefore, the number of ports has the greatest effect on the computation time. Throughout the experiments on the above 20 problem instances, it can be seen that the solution found in the second stage reduces the objective function value about 4%. However, as we mentioned above, the second stage takes too much computation time. Hence, if the problem size is larger than those problem instances, only the first stage should be used and we may lose about 5% in comparison with optimal solution.

Table 4 Comparison between MIP, the LP-based GA, and the hybrid GA for 20 problems

Problem instance	Computation time (s)			Penalty (%) of the Hybrid GA		
	MIP	LP-based GA	Hybrid GA	vs. Lower Bound	vs. MIP	vs. LP-based GA
1	–	–	1,221	3.10	–	–
2	–	–	14,159	4.11	–	–
3	442	34,076	589	2.07	1.96	0.42
4	182	19,166	358	2.07	1.94	0.22
5	–	–	7,559	4.29	–	–
6	–	–	12,072	2.66	–	–
7	–	–	34,603	2.78	–	–
8	27,646	37,001	745	1.66	1.54	0.13
9	–	–	1,987	2.61	–	–
10	–	–	1,396	2.03	–	–
11	–	–	2,643	2.53	–	–
12	18,333	17,132	377	1.57	1.45	0.15
13	–	–	2,231	2.25	–	–
14	–	–	3,837	2.44	–	–
15	–	–	10,129	2.59	–	–
16	–	38,735	686	2.07	–	0.20
17	–	–	1,084	2.01	–	–
18	–	–	–	–	–	–
19	–	–	1,513	2.06	–	–
20	–	–	5,058	2.12	–	–

5 Conclusions

In this paper, we considered the problem of empty container positioning with respect to leasing and purchasing. The problem considers many decision variables, such as the number of empty containers to be positioned, the number of leased and purchased containers, and the number of leased containers to be returned, and the solution is useful to a shipping company for decision making at an operational level. MIP was proposed for this problem and two GAs were suggested. The results showed that the LP-based GA and the hybrid GA are capable of solving problems of larger size. Moreover, through experiments, it was demonstrated that the hybrid GA is more efficient than the LP-based GA in terms of computation time. The numerical experiments also pointed out the limitation of the hybrid GA when the number of ports becomes too large.

To solve a very large problem, we should reduce the size of the LP problem. Assumptions may be given to reduce the number of continuous variables and constraints. The returning of leased containers is an example. If it is assumed that the leased containers can be returned at any port, then the index k can be omitted. Note that k appears in many variables. These are x_{ijvt}^{elk} , x_{ijvt}^{flk} , w_{ivt}^k , and LI_{ivt}^k . Therefore, omitting k can

considerably reduce the number of variables. It would also remove constraints relating to this index [Constraints (2a) and (6)]. Another assumption is not to consider the vessel management. With this assumption, variables and constraints that relate to vessel transportation are omitted, such as f_{ijvt}^1 , f_{ijvt}^2 , s_{ijt}^1 , s_{ijt}^2 , and Constraints (7) and (8). However, these assumptions make the problem less meaningful and impractical. A better approach would be to study other heuristic methods that can lead to a feasible solution for the problem, and this will be the subject of further research.

Acknowledgments The authors are grateful to the careful and constructive reviews from the guest editor (Professor Herbert Kopfer) and two anonymous referees. The authors also thank Mr. Fischer at Softship Data Processing Pte. Ltd. for providing practical advices on parameters for the mathematical model. This work was supported by the Grant of the Korean Ministry of Education, Science and Technology (The Regional Core Research Program/Institute of Logistics Information Technology).

References

- Crainic TG, Dejax P, Delorme L (1989) Models for multimode multicommodity location problems with interdepot balancing requirements. *Ann Oper Res* 18:279–302
- Crainic TG, Delorme L (1993) Dual-ascent procedures for multicommodity location–allocation problems with balancing requirements. *Transport Sci* 27:90–101
- Crainic TG, Gendreau M, Soriano P, Toulouse M (1993) A Tabu search procedure for multicommodity location/allocation with balancing requirements. *Ann Oper Res* 41:359–383
- Gen M, Cheng R (2000) *Genetic algorithms and engineering optimization*. Wiley, New York
- Gendron B, Crainic TG (1997) A parallel branch-and-bound algorithm for multicommodity location with balancing requirements. *Comput Oper Res* 24:829–847
- Gendron B, Potvin JY, Soriano P (2003) A Tabu search with slope scaling for the multicommodity capacitated location problem with balancing requirements. *Ann Oper Res* 122:193–217
- Goldberg D (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, USA
- Li JA, Liu K, Leung SCH, Lai KK (2004) Empty container management in a port with long-run average criterion. *Math Comput Model* 40:85–100
- Li JA, Leung SCH, Wu Y, Liu K (2007) Allocation of empty containers between multi-ports. *Eur J Oper Res* 182:400–412
- Lou YC, Guinard M, Chen CH (2001) A hybrid approach for integer programming combining genetic algorithms, linear programming and ordinal optimization. *J Intel Manufact* 12:509–919
- Nemhauser GL, Wolsey LA (1988) *Integer and combinatorial optimization*. Wiley, USA
- Pedroso JP (1998) An evolutionary solver for linear integer programming. BSIS Technical Report No. 98–7
- Rajasekharan M, Peters BA, Yang T (1998) A genetic algorithm for facility design in flexible manufacturing system. *Int J Prod Res* 36:95–110
- Shen WS, Khoong CM (1995) A DSS for empty container distribution planning. *Decis Support Syst* 15:75–82