Computers & Industrial Engineering 108 (2017) 149–164

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Heuristic algorithm for the container loading problem with multiple constraints

Liu Sheng [a,*], Shang Xiuqin [a], Cheng Changjian [b], Zhao Hongxia [a], Shen Dayong [c], Wang Feiyue [a]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China
[b] Institute of Process Engineering, Chinese Academy of Sciences, 100190 Beijing, China
[c] Qingdao Academy of Intelligent Industries, 266000 Qingdao, Shandong, China

ARTICLE INFO

Article history:
Received 9 January 2016
Received in revised form 6 March 2017
Accepted 13 April 2017
Available online 21 April 2017

Keywords:
Packing
Container loading
Complete shipment constraint
Shipment priority
Simulated annealing

ABSTRACT

This paper addresses the container loading problem with multiple constraints that occur at many manufacturing sites, such as furniture factories, appliances factories, and kitchenware factories. These factories receive daily orders with expiration dates, and each order consists of one or more items. On a particular day, certain orders expire, and the expiring orders must be handled (shipped) prior to the non-expiring ones. All of the items in an order must be placed in one container, and the volume of the container should be maximally utilized. A heuristic algorithm is proposed to standardize the packing of (order) items into a container. The algorithm chooses the expiring orders first before handling the non-expiring orders. In both steps, the algorithm first selects a collection of orders by considering a simulated annealing strategy and subsequently packs the collection of orders into the container via a tree-graph search procedure. The validity of the algorithm is examined through experimental results using BR instances.

1. Introduction

Many factories, such as furniture factories, appliance factories, and kitchenware factories, are confronted with the problem of packing item orders. Every day, these factories receive orders from buyers worldwide and ship the items to the buyers in containers. The factories want to load as many items as possible into a container to reduce shipping costs. Each order consists of one or more different items. On any given day, certain orders expire; the expiring orders must be packed and shipped prior to the non-expiring ones. If one item of an order is packed in the container, all items of the order must be packed in the same container. If one item of an order is not packed in the container, all items of the order must not be packed in the container. To protect the items, each item must be loaded with its height parallel to the height of the container. The bottom of each item must be supported by the container floor or by the top of a single item to simplify the unloading process and to ensure the stability of the items.

According to the typology proposed by Dyckhoff (1990), this issue is a 3/B/O problem. According to the recent typology proposed by Wäscher, Haußner, and Schumann (2007), this issue is a three-dimensional single knapsack problem (**3D-SKP**) or a three-dimensional single large object placement problem (3D-SLOPP). Following the review of the paper by Bortfeldt and Wäscher (2013), **3D-SKP** in this paper is a three-dimensional, container-loading problem (**3D-CLP**) with the orientation constraint (**C1**), the stability constraint (**C2**), the guillotine-cutting constraint (**C3**), the complete-shipment constraint (**C4**), and the loading priority constraint (**C5**).

The rest of the paper is organized as follows: In Section 2, we provide a literature review of three-dimensional container-loading problems. In Section 3, we review the methods that solve the container loading problem of item orders. In Section 4, we analyse the results of the presented method based on simulations created from BR instances (Bischoff & Ratcliff, 1995; Davies & Bischoff, 1999). Finally, in Section 5, we present the paper's conclusions perspectives for future research.

2. Literature review

Since the seminal work by George and Robinson (1980), **3D-CLP** has received increasing attention from academic researchers and

* Corresponding author.
E-mail addresses: liusheng7801@163.com, sheng.liu@ia.ac.cn (L. Sheng), xiuqin.zhao@ia.ac.cn (S. Xiuqin), changjian.cheng@ia.ac.cn (C. Changjian), hongxia.zhao@ia.ac.cn (Z. Hongxia), dayong.shen@qaii.ac.cn (S. Dayong), feiyuewang@gmail.com (W. Feiyue).

http://dx.doi.org/10.1016/j.cie.2017.04.021
0360-8352/

industry professionals. Bischoff and Ratcliff (1995) provide an excellent overview of the practical requirements that may be imposed on this problem. Bortfeldt and Wäscher (2013) provide a comprehensive scheme to categorize the constraints of loading a container and determine that the existing approaches have limited practical value because they do not sufficiently address the constraints encountered in practice. In this paper, we discuss **3D-CLP** with several practical constraints (described in the Introduction section). To the best of our knowledge, there are no published approaches that address **3D-CLP** using the practical constraints we investigate. We will briefly classify and discuss several of the recent advances in this problem.

3D-CLP is considered an NP-hard (Bischoff & Marriott, 1990) problem. Exact algorithms are usually constrained by a situation called the combinatorial space explosion when the number of item types increases. Consequently, notably few exact algorithms exist.

Martello et al. (2000); also see Den Boef et al., 2005) present a branch-and-bound algorithm that addresses the single knapsack problem and—based on this algorithm—a branch-and-bound method for addressing the single bin-size, bin-packing problem. The orientation of all of the items are fixed, and no further constraints are considered.

Hifi (2004) introduces an exact depth-first search and a dynamic programming algorithm for solving the 3D SLOPP in a cutting context. The orientation and the guillotine cutting constraint are both considered. The number of items per type is unlimited.

Fekete, Schepers, and Van der Veen (2007) develop an exact algorithm for higher-dimensional orthogonal packing problems, where the small items have no fixed orientations, and no other constraints are considered.

Heuristic methods prove to be a more realistic alternative for addressing **3D-CLP**. Although heuristic methods may only find suboptimal solutions, they can produce sufficiently good solutions in a reasonable timeframe.

Orientation and stability constraints are the most studied constraints in the literature. In **3D-CLP**, small items may have at most six orthogonal orientations in the container. Pisinger (2002) and Egeblad and Pisinger (2009) assume that the small items may be rotated in any orthogonal direction. Most heuristic methods (e.g., Bischoff, Janetz, & Ratcliff, 1995; Bortfeldt & Gehring, 2001; Fanslau & Bortfeldt, 2010; Gehring & Bortfeldt, 2002; He & Huang, 2010, 2011; Huang & He, 2009; Lim, Ma, Xu, & Zhang, 2012; Lim, Rodrigues, & Wang, 2003; Moura & Oliveira, 2005; Parreño, Alvarez-Valdés, Tamarit, & Oliveira, 2008; Zhu, Huang, & Lim, 2012; Zhu & Lim, 2012) assume that some orientations are forbidden.

Load stability is often considered the most important issue after container space utilization in the literature (e.g., Bischoff & Ratcliff, 1995; Bortfeldt, Gehring, & Mack, 2003; Eley, 2002; Fanslau & Bortfeldt, 2010; Gehring & Bortfeldt, 2002; Liu, Tan, Xu, & Liu, 2014; Ren, Tian, & Sawaragi, 2011; Zhang, Peng, & Leung, 2012; Zhu & Lim, 2012; Zhu et al., 2012). Fanslau and Bortfeldt (2010) and Zhu et al. (2012) consider two situations where small items are fully or partially supported. Partial support is required by Gehring and Bortfeldt (1997) and Mack, Bortfeldt, and Gehring (2004).

The guillotine cutting constraint is often viewed from a loading perspective. A guillotine pattern represents a type of loading pattern that can be packed easily. A loading pattern is said to be *guillotineable* if it can be obtained by a series of "cuts" parallel to the container faces (especially the vertical faces). The guillotine cutting constraint is considered in Hifi (2002), Morabito and Arenalest (1994), and Liu et al. (2014). The loading pattern obtained by Pisinger (2002) is guillotineable although it is not declared as such.

In practice, the available container space is not sufficiently large to accommodate all small items, and the loading of some items may be more desirable than the loading of others. Thus, shipment priorities (Bortfeldt & Wäscher, 2013, also called loading priorities by Bischoff & Ratcliff, 1995) exist for some items. The shipment priority constraint is considered by Bortfeldt and Gehring (1999), Ren et al. (2011), and Wang, Lim, and Zhu (2013).

Certain subsets of loaded items may include functional or administrative supplies (Bischoff & Ratcliff, 1995). If one item of a subset is loaded, all other items of that subset must also be loaded. If one item cannot be loaded, no item of the subset will be loaded at all. Two cases can be distinguished: In the first case, all items of a subset must be included in the shipment. In the second case, all items of a subset have to be loaded into the same container. Eley (2003) considers the first case in a multiple heterogeneous large object placement problem.

A heuristic algorithm for container-loading of furniture, by Egeblad, Garavelli, Lisi, and Pisinger (2010), is remarkable in that a large variety of irregular items are considered and many practical constraints are satisfied. However, the loading priority constraint and the complete-shipment constraint are not considered. Lim, Ma, Qiu, and Zhu (2013) consider the axle-weight constraint when solving the single container loading problem. Chen, Lee, and Shen (1995) provide an analytical model for the container loading problem. Junqueira, Morabito, and Sato Yamashita (2012) propose MIP-based approaches for the container loading problem with multi-drop constraints. Liu, Zhao, Dong, and Cheng (2016) present a heuristic algorithm for container loading of pallets with infill boxes.

The great majority of the methods mentioned above obey the orientation constraint and the stability constraint. Some methods also include additional constraints, e.g., such as the guillotine constraint, the loading priority constraint and the complete-shipment constraint. However, no approaches simultaneously consider all of the five constraints. Thus, we will develop a new algorithm that can handle the container loading problem with all of the five constraints.

3. Method for the container loading problem of item orders

3.1. Problem definition

The **3D-CLP** in this paper is called the container-loading problem with multiple constraints **CLPMC**. **CLPMC** is defined as follows.

A factory has m unfinished orders that are set to expire and n unfinished orders that do not expire. The total number of items in the $m+n$ orders is k. The k items are characterized by lengths $(l_1, l_2, \ldots, l_k)$, widths $(w_1, w_2, \ldots, w_k)$, and heights $(h_1, h_2, \ldots, h_k)$. The container C is characterized by the length L, the width W and the height H. The objective is to load a subset of these $m+n$ orders with maximum item volume into C. Additionally, the five constraints (**C1**-**C5**) must be fulfilled.

We define:

$$o_{ij} = \begin{cases} 1; & \text{if the } j\text{th item is in the } i\text{th expiring order}, \\ 0; & \text{otherwise}. \end{cases} \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, k), \tag{1}$$

$$r_{ij} = \begin{cases} 1; & \text{if the } j\text{th item is in the } i\text{th unexpiring order}, \\ 0; & \text{otherwise}. \end{cases} \quad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k), \tag{2}$$

$$a_i = \begin{cases} 1; & \text{if the items of the } i\text{th expiring order is packed in the container}, \\ 0; & \text{otherwise}. \end{cases} \quad (i = 1, 2, \ldots, m), \tag{3}$$

and

$$b_i = \begin{cases} 1; & \text{if the items of the } i\text{th unexpiring order is packed in the container,} \\ 0; & \text{otherwise.} \end{cases} \quad (i = 1, 2, \ldots, n). \tag{4}$$

Clearly, the conditions defined by Eqs. (5)–(7) should be satisfied by the CLPMC. Eq. (5) means that all items are included in the orders. Eq. (6) defines that each item is only included in one order. Eq. (7) means that the total volume of the loaded items is not greater than the volume of the container.

$$\sum_{i=1}^{m}\sum_{j=1}^{k} o_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{k} r_{ij} = k, \tag{5}$$

$$\sum_{i=1}^{m} o_{ij} + \sum_{i=1}^{n} r_{ij} = 1 \quad (j = 1, 2, \ldots, k), \tag{6}$$

and

$$\sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j + \sum_{i=1}^{n}\sum_{j=1}^{k} b_i r_{ij} l_j w_j h_j \leqslant L * W * H. \tag{7}$$

The solution of CLPMC can be obtained by first solving the integer programming (IP) problem (**IP_1**):

$$\max\left\{ v_O = \sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j \right\} \tag{8}$$

Subject to Eqs. (5), (6) and

$$\sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j \leqslant L * W * H \tag{9}$$

and then solving the IP problem (**IP_2**):

$$\max\left\{ v_R = \sum_{i=1}^{n}\sum_{j=1}^{k} b_i r_{ij} l_j w_j h_j \right\} \tag{10}$$

Subject to Eqs. (5), (6) and

$$\sum_{i=1}^{n}\sum_{j=1}^{k} b_i r_{ij} l_j w_j h_j \leqslant L * W * H - \sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j \tag{11}$$

In the situation where all five constraints (**C1-C5**) are imposed, $\{a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n\}$ is a feasible solution. A solution is also called a **container loading plan** in this paper. **IP_1** describes a criterion to load the expiring orders whereas **IP_2** defines a criterion to load the non-expiring orders. The constraint (C4) is satisfied by using a_i (i = 1, 2, …, m) and b_j (i = 1, 2, …, n) as the decision variables in **IP_1** and **IP_2**. The constraint (C5) is fulfilled by solving **IP_1** before solving **IP_2**. The constraints (C1) and (C2) are met when we construct the **container loading plan.** We use a wall-building strategy to construct the **container loading plan.** Consequently, the **container loading plan** may meet the constraint (C3) (Liu et al., 2014).

3.2. The overall algorithm of CLPMC

We refer here to our heuristic algorithm for **CLPMC** as **HCLPMC** (heuristic algorithm for the container loading problem with multiple constraints). To solve the container loading problem with the constraints (**C1-C5**), **HCLPMC** uses a four-step strategy. The flow diagram of the strategy is illustrated in Fig. 1. The first step is to select a subset O' from the expiring orders O by solving a one-dimensional knapsack problem (**KP_1D_O**):

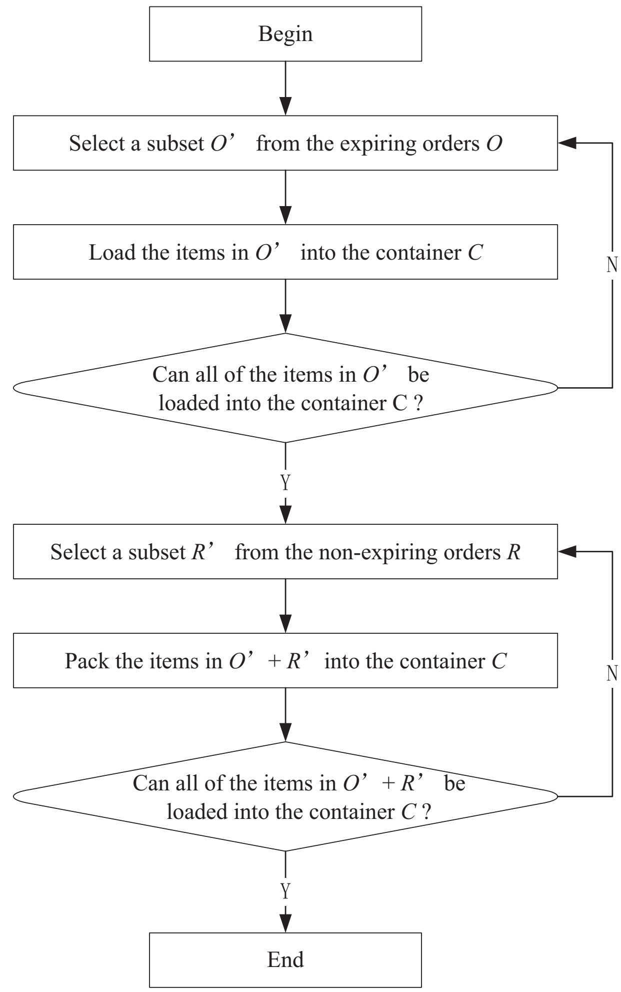

Fig. 1. The flow diagram of **HCLPMC**.

$$\max\left\{ v_O = \sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j \right\} \tag{12}$$

Subject to

$$\sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j \leqslant L * W * H - V_\Delta \tag{13}$$

where $\{a_1, a_2, \ldots, a_m\}$ is a feasible solution and V_Δ is a parameter whose initial value is 0. The second step is to load the items of O' into container C by solving a container loading problem (**CLP**). If the items of O' cannot be entirely placed in C, let $V_\Delta = V_\Delta + L * W * H/200$ and solve **KP_1D_O** again to update O'. The third step is to select a subset R' from the non-expiring orders R by solving a one-dimensional knapsack problem (**KP_1D_R**):

$$\max\left\{ v_R = \sum_{i=1}^{m}\sum_{j=1}^{k} b_i r_{ij} l_j w_j h_j \right\} \tag{14}$$

Subject to

$$\sum_{i=1}^{m}\sum_{j=1}^{k} b_i r_{ij} l_j w_j h_j \leqslant L * W * H - \sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j - V'_\Delta \tag{15}$$

where $\{b_1, b_2, \ldots, b_m\}$ is a feasible solution and V'_Δ is a parameter whose initial value is 0. The fourth step is to pack the items in O' and R' into C. If the items in O' and R' cannot be entirely placed in

C, let $V'_\Delta = V'_\Delta + \left(L * W * H - \sum_{i=1}^{m}\sum_{j=1}^{k} a_i o_{ij} l_j w_j h_j\right)/200$ and solve **KP_1D_R** again to update R'. With the four steps, **HCLPMC** can obtain a container loading plan that satisfies the constraints (**C1**-**C5**).

The overall algorithm **HCLPMC** (Procedure 1) invokes two procedures: **LOAD_EXP** (Procedure 2) and **LOAD_ALL** (Procedure 3). **LOAD_EXP** loads the expiring orders, while **LOAD_ALL** loads all of the orders. **LOAD_EXP** (Procedure 2) and **LOAD_ALL** (Procedure 3) each invoke two procedures: **SA_KP_1D** (Procedure 4) and **TS_CLP** (Procedure 5). **SA_KP_1D** uses simulated annealing to solve **KP_1D_O** and **KP_1D_R**, while **TS_CLP** packs items into the container C using a tree search algorithm. This tree search algorithm satisfies the constraints (**C1**), (**C2**) and (**C3**), simultaneously. Using the five labels, L, W, H, P, O, and R to represent the length, width, and height of the container, the set of items, the set of expiring orders, and the set of non-expiring orders, respectively.

In **HCLPMC**, **LOAD_EXP** is invoked to pack a subset of O into the container. Next, **LOAD_ALL** is invoked to load the subset of O and a subset of R into the container. Thus, the solution *clp* is obtained.

Procedure 1.

```
HCLPMC(L,W,H,O,R)
  zlet clpo and clp denote two container loading plans
  clpo := LOAD_EXP(L,W,H,O)
  clp := LOAD_ALL(L,W,H,R,clpo)
  return clp
```

LOAD_EXP is used to fill the container with the items of the expiring orders. A step-down strategy is used here. First, we obtain the upper bound by applying **TS_CLP** with items from O. If the upper bound contains all of the items in O, the upper bound is the solution of **LOAD_EXP**. Otherwise, using (the total item volume in the upper bound – 1) as the weight limit, we solve a knapsack problem by invoking **SA_KP_1D** to obtain a new item set I. Then, **TS_CLP** is used again to obtain a new solution *clpo*. If *clpo* contains the items found in I, *clpo* is the best solution. Otherwise, we will circularly invoke **SA_KP_1D** with (the total item volume in the new solution – 1) as the weight limit and invoke **TS_CLP** until the ultimate solution is obtained.

Procedure 2.

```
LOAD_EXP (L,W,H,O)
  let clpo and I denote a container loading plan and an item
   set
  let wei_lim denote the weight limit of the knapsack
  I := all the items in O
  clpo := TS_CLP(L,W,H,I)
  while clpo does not contain all the items in I
    wei_lim := the total item volume in clpo −1
    I := SA_KP_1D(O,wei_lim)
    clpo := TS_CLP(L,W,H,I)
  return clpo
```

LOAD_ALL tries to fill the container with the non-expiring items and the items that are loaded by **LOAD_EXP**. The parameter *clpo* is the container loading plan obtained by **LOAD_EXP**. First, we try to pack all of the items in R and *clpo* into the container. If all of these items are packed into the container, the obtained container loading plan *clp* is the ultimate solution of **LOAD_ALL**. Otherwise, using the total item volume of R in *clp* – 1 as the weight limit, we solve a knapsack problem by invoking **SA_KP_1D** to obtain a new item set. Then, we pack the items in the new set and *clpo* into the container. If all of these items are completely packed in the container, we obtain the ultimate solution. Otherwise, we invoke **SA_KP_1D** and **TS_CLP** recursively until the ultimate solution is obtained.

Procedure 3.

```
LOAD_ALL (L,W,H,R,clpo)
  let clp and I denote a container loading plan and an item set
  let wei_lim denote the weight limit of the knapsack
  I := all the items in R and clpo
  clp := TS_CLP(L,W,H,1)
  while clp does not contain all the items in I
    wei_lim := the total volume of ((the items in clp) ∩ (the
   items in R)) −1
    I := SA_KP_1D (R,wei_lim)
    I := I + the items in clpo
    clp := TS_CLP(L,W,H,l)
  return clp
```

In **SA_KP_1D**, OR is an expiring (or non-expiring) order set where *wei_lim* is an integer. **SA_KP_1D** tries to obtain a subset of OR with a maximum volume that is no more than *wei_lim*. This is a one-dimensional knapsack problem. In the experiments, we find that the dynamic programming method cannot obtain results within acceptable times since *wei_lim* is so large that it is time-consuming to create an $N \times wei\_lim$ table. To solve this one-dimensional knapsack problem, **SA_KP_1D** uses a simulated annealing algorithm instead of a dynamic programming method. We repeat the annealing loop for 20 iterations (specified by experience). The adjacent annealing processes cannot produce a better solution via more iterations. A less precise solution is accepted with a maximum probability of 0.5 in each annealing iteration step. Based on our experience, the initial temperature of 0.998^{-N^2}, the end temperature of 1, and the reducing rate of 0.998, our simulated-annealing-based procedure may produce a relatively good solution within an acceptable time frame.

Procedure 4..

```
SA_KP_1 D (OR,wei_lim)
  let O′, o and temp be an empty order set, an order and a
   temperature value
  let V () be the function to compute the total volume of items
  let N be the number of orders in OR
  temp := 0.998^(−N^2)
  while a better solution is obtained in 20 adjacent loops
    while temp > 1
      randomly select an order o in OR
      if ((O′ doesn't contain o) and (V (O′) + V (o) ≤ wei_lim))
        insert o into O′
      else if (O′ contains o)
        remove o from O′ with the probability
   0.5e^(V(O)(0.998N^2−1/temp))
          //0.5 is the maximum value of the probability
      temp := temp ∗ 0.998
    temp := 0.998^(−N^2)
  return the items in O′
```

TS_CLP is a tree search procedure based on a wall-building strategy. As shown in Fig. 2, a layer that is parallel to the *x*-*z* plane is called a x-layer, and a layer that is parallel to the *y*-*z* plane is

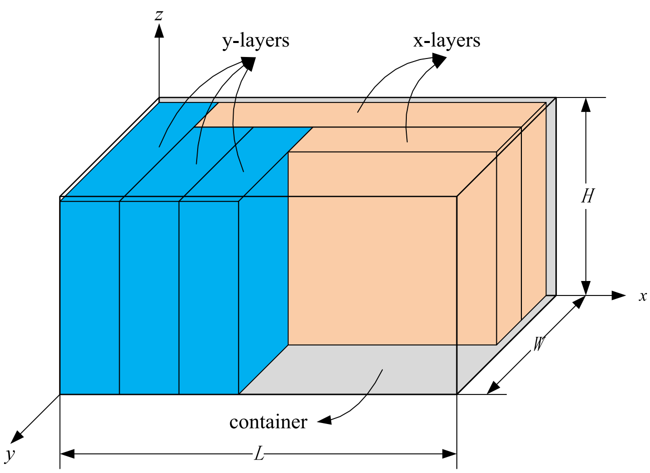

Fig. 2. An container loading diagram with five layers.

called a y-layer. For each tree node in the ternary tree, **TS_CLP** creates a set of candidate x-layers and selects the x-layers with the highest filling rates in the set as the first child, then creates a set of candidate y-layers and select the y-layer with the highest filling rate in the set as the second child. If the filling rate of the x-layer and the second highest filling rate is higher than the filling rate of the y-layer when compared to the highest filling rate, the x-layer will be selected as the third child. If the filling rate of the y-layer with the second highest filling rate is higher than the filling rate of the x-layer with the highest filling rate, the y-layer will be selected as the fourth child. It is obvious that the third child and the fourth child cannot exist simultaneously. Thus, each tree node has three children at most.

Procedure 5.

TS_CLP(L,W,H,I)
  //FR(l) means the filling rate of the layer l.
  // LI, $L2$, $L3$ and $L4$ are 4 layer sets
  create and arrange candidate x layers $xl_x, xl_2, \ldots$, from I such that FR (xl_1) $\geq$ FR (xl_2) $\geq \cdots$
  create and arrange candidate y layers $yl_x, yl_2, \ldots$, from I such that FR(yl_1) $\geq$ FR(yl_2) $\geq \cdots$
  $L1 := xl_1$ + TS_CLP(L, W – the thickness of xl_1, H, I – the items in xl_1)
  $L2 := yl_1$ + TS_CLP(L – the thickness yl_1, W, H, I – the items myl_1)
  if(FR(xl_2) > FR(yl_1))
    $L3 := xl_2$ + TS_CLP(L, W – the thickness of xl_2, H, I – the items in xl_2)
  if(FR(yl_2) > FR(xl_1))
    $L4 := yl_2$ + TS_CLP(L – the thickness of yl_2, W, H, I – the items in yl_2)
  return the one in {LI,$L2$,$L3$,$L4$} with the highest box volume

4. Computational experiments and results

HCLPMC is implemented in C# and run on a server with an Intel Core i7 870 @2.93 GHz processor and Microsoft Windows 7 Professional operating system. The compiling environment is Microsoft Visual Studio 2010.

Since **TS_CLP** (Procedure 5) is an algorithm for the container loading problem with the constraints **C1**-**C3**, we first use BR1-BR15 (Bischoff & Ratcliff, 1995; Davies & Bischoff, 1999) to test **TS_CLP** (Procedure 5) and compare **TS_CLP** with some recent algorithms that satisfy the constraints **C1** and **C2**. The numbers of item types in BR1-BR15 are 3, 5, 8, 10, 12, 15, 20, 30, 40, 50, 60, 70, 80, 90, and 100, respectively. Each item type includes at least one item. The item sets in BR1 to BR15 vary from homogeneous through weakly heterogeneous to strongly heterogeneous. Usually, the cases in BR1-BR7 are considered weakly heterogeneous loading problems, while the cases in BR8-BR15 are considered strongly heterogeneous ones. The container used in the BR problems is 587 cm long, 233 cm wide, and 220 cm high.

The instances from BR1 to BR7 are tested in H_BR (by Bischoff & Ratcliff, 1995), GA_GB (by Gehring & Bortfeldt, 1997), PTSA (by Bortfeldt et al., 2003), GRASP (by Moura & Oliveira, 2005), MFB (by Lim, Rodrigues, & Yang, 2005), RHA (by Juraitis, Stonys, Starinskas, Jankauskas, & Rubliauskas, 2006), H_B (by Bischoff, 2006) and SPBBL-CC4 (by Bortfeldt & Mack, 2007) which are introduced into the weakly heterogeneous loading problem.

MSA (by Parreno, Alvarez-Valdes, Oliveira, & Tamarit, 2007), HSA (by Zhang, Peng, Zhu, & Chen, 2009), VNS (by Parreno, Alvarez-Valdes, Oliveira, & Tamarit, 2010), CLTRS (by Fanslau & Bortfeldt, 2010), FDA (by He & Huang, 2011), ID-GLTS (by Zhu & Lim, 2012), and HBMLS (by Zhang et al., 2012) test all the instances from BR1 to BR15 while A2 (by Huang & He, 2009) tests the ones from BR8 to BR15.

Table 1 reports the platforms for testing TS_CLP and other algorithms. The data regarding other algorithms are from the cited literature at the beginning of this section. Some algorithms also describe the platforms used in detail, while others did not reveal this information.

Table 2 reports the computational results of TS_CLP and other algorithms for the instances from BR1 to BR7. All the data in Table 2 denote the average filling rate (%) for one case. From Table 2, we can find that TS_CLP is worse than CLTRS, ID_GLTS and HBMLS for BR1-BR7. The reason is that the guillotine cutting constraint is not considered in CLTRS, ID_GLTS or HBMLS.

Table 3 reports the computational results of TS_CLP and other algorithms for BR8-BR15. All the data in Table 3 denotes the

Table 1
Platforms for testing TS_CLP and other Algorithms.

Algorithm	Platform
H_BR (Bischoff & Ratcliff, 1995) (Heuristic by Bischoff and Ratcliff)	–
GA_GB (Gehring & Bortfeldt, 1997) (Genetic algorithm by Gehring and Bortfeldt)	Pentium 130
PTSA (Bortfeldt et al., 2003) (Parallel tabu search algorithm)	Pentium 2 GHz
GRASP (Moura & Oliveira, 2005) (Greedy randomized adaptive search procedure)	–
MSA (Parreno et al., 2007) (Maximal-space algorithm)	Pentium Mobile 1500Mhz, 512 MB Ram, C++
HSA (Zhang et al., 2009) (Hybrid simulated annealing)	Core 2 Duo 2.0 GHz, C++
A2 (Huang & He, 2009) (Caving degree approach)	1.7 GHz, Windows, Java
VNS (Parreno et al., 2010) (Variable Neighborhood Search)	Pentium Mobile 1500Mhz, 512 MB Ram, C++
CLTRS (Fanslau & Bortfeldt, 2010) (Container loading by tree search)	Set A: 2.6 GHz; set B: 800 MHz
FDA (He & Huang, 2011) (Fit degree approach)	Xeon 2.33 GHz, Java, J2SE V1.5.0_14
HBMLS (Zhang et al., 2012) (heuristic block loading algorithm based on multi-layer search)	Xeon X5460@3.16 GHz, Debian Linux, C++, gcc 4.3.2
ID-GLTS (Zhu & Lim, 2012) (Iterative-Doubling Greedy–Lookahead Tree Search)	Xeon E5520@2.27 GHz, 8G RAM. Linux, Java, 1.6.0
TS_CLP (Tree search for container loading problem)	Core i7 870 @2.93 GHz, Windows 7, C#, visual studio 2010

Table 2
Results of TS_CLP and Other Methods for BR1-BR7.

Algorithm	Constraint	Filling rate (%)						
		BR1	BR2	BR3	BR4	BR5	BR6	BR7
H_BR	C1&C2	85.40	86.25	85.86	85.08	85.21	83.84	82.95
GA_GB	C1&C2	85.80	87.26	88.10	88.04	87.86	87.85	87.68
PTSA	C1&C2	93.52	93.77	93.58	93.05	92.34	91.72	90.55
GRASP	C1	93.52	93.77	93.58	93.05	92.34	91.72	90.55
MSA	C1	93.85	94.22	94.25	94.09	93.87	93.52	92.94
HSA	C1&C2	93.81	93.94	93.86	93.57	93.22	92.72	91.99
VNS	C1	94.93	95.19	94.99	94.71	94.33	94.04	93.53
CLTRS	C1	95.05	95.39	95.45	95.18	94.96	94.80	94.26
	C1&C2	94.50	94.67	94.74	94.41	94.05	93.83	93.15
FDA	C1	92.92	93.93	93.71	93.68	93.73	93.63	93.14
HBMLS	C1	94.92	95.48	95.69	95.53	95.44	95.38	95.00
	C1&C2	94.43	94.87	95.06	94.89	94.68	94.53	93.96
ID-GLTS	C1	95.59	96.13	96.30	96.15	95.98	95.81	95.36
	C1&C2	94.40	94.85	95.10	94.81	94.52	94.33	93.59
TS_CLP	C1, C2&C3	90.62	91.51	92.43	92.35	92.45	92.37	92.13

Table 3
Results of TS_CLP and Other Methods for BR8-BR15.

Algorithm	Constraint	Filling rate (%)							
		BR8	BR9	BR10	BR11	BR12	BR13	BR14	BR15
MSA	C1	91.02	90.46	89.87	89.36	89.03	88.56	88.46	88.36
HSA	C1&C2	90.56	89.7	89.06	88.18	87.73	86.97	86.16	85.44
A2	C1	88.41	88.14	87.9	87.88	87.92	87.92	87.82	87.73
VNS	C1	92.78	92.19	91.92	91.46	91.2	91.11	90.64	90.38
CLTRS	C1	93.70	93.44	93.09	92.81	92.73	92.46	92.40	92.40
	C1&C2	92.26	91.48	90.86	90.11	89.51	88.98	88.26	87.57
FDA	C1	92.92	92.49	92.24	91.91	91.83	91.56	91.3	91.02
HBMLS	C1	94.66	94.30	94.11	93.87	93.67	93.45	93.34	93.14
	C1&C2	93.27	92.60	92.05	91.46	90.91	90.43	89.80	89.24
ID-GLTS	C1	94.80	94.53	94.35	94.14	94.10	93.86	93.83	93.78
	C1&C2	92.65	92.11	91.60	90.64	90.35	89.69	89.07	88.36
TS_CLP	C1, C2&C3	91.95	91.64	91.42	91.14	**90.98**	**90.60**	**90.27**	**89.84**

average filling rate (%) for one case. For BR8-BR11, TS_CLP is worse than HBMLS and ID_GLTS. Compared to HBMLS, TS_CLP improves performance by 0.07%, 0.17%, 0.47% and 0.6% for BR12-BR15, respectively.

The mean filling rates in CLTRS, ID-GLTS, HBMLS and TS_CLP for BR1-BR15 are 91.89%, 92.40%, 92.81%, and 91.44%, respectively. Obviously TS_CLP is weaker than CLTRS, ID-GLTS and HBMLS in most case. Table 2 and Table 3 show that CLTRS, ID-GLTS, HBMLS obtain better results than TS_CLP on BR1-BR11, but obtain worse results than TS_CLP on BR12-BR15. Thus, we can say TS_CLP is more suitable for strongly heterogeneous situations instead of weakly heterogeneous instances. In addition, TS_CLP obeys the guillotine cutting constraint. Therefore, TS_CLP can be used in the context in which the guillotine cutting constraint is mandatory.

Tables 4 and 5 reports the computation times of TS_CLP and other algorithms for BR1-BR7 and BR8-BR15, respectively. From the two tables, we find the computation times of TS_CLP for BR1-BR15 are reasonable.

Due to the lack of test data for **CLPMC**, we generated 1500 instances from the 1500 instances of BR1-BR15 (Bischoff & Ratcliff, 1995; Davies & Bischoff, 1999) by grouping items into orders and adding loading priorities into the different orders. Accordingly, we call our case groups BR1v, BR2v, ..., BR15v.

The items $b_1, b_2, \ldots, b_i, \ldots$ in each case of BR1-BR15 are associated with a sequence of orders $or_1, or_2, \ldots, or_j, \ldots$ for each case of BR1v-BR15v. The quantities of items in or_j is Mod(j, 4) (if Mod (j, 4) $\neq 0$) or 4 (if Mod(j, 4) = 0). Fig. 3 shows the relationships between the items and the orders in each case. Occasionally, if

Table 4
Computation Times of TS_CLP and Other Methods for BR1-BR7.

Algorithm	Constraint	Computation time (s)							
		BR1	BR2	BR3	BR4	BR5	BR6	BR7	Mean
PTSA	C1&C2	36	48	97	138	179	150	198	
MSA	C1	1.27	2.32	4.62	6.52	8.58	12.23	19.25	
HSA	C1&C2	20.33	35.68	59.00	75.05	80.63	88.89	101.52	
VNS	C1	2.98	5.60	11.09	15.12	22.62	31.71	58.00	
CLTRS	C1	–	–	–	–	–	–	–	52
	C1&C2	–	–	–	–	–	–	–	320
FDA	C1	1.16	2.54	5.14	7.66	10.38	16.66	29.54	
ID-GLTS	C1	–	–	–	–	–	–	–	503.45(BR1-BR15)
	C1&C2	–	–	–	–	–	–	–	150.55(BR1-BR15)
HBMLS	C1	14.1	34.18	79.43	115.59	155.1	217.57	327.88	
	C1&C2	14.71	36.43	80.33	116.13	153.38	204.15	295.69	
TS_CLP	C1, C2&C3	105.6	117.6	125.3	124.0	119.4	132.7	152.2	125.3

Table 5
Computation Times of TS_CLP and Other Methods for BR8-BR15.

Algorithm	Constraint	Computation time (s)								
		BR8	BR9	BR10	BR11	BR12	BR13	BR14	BR15	Mean
MSA	C1	38.20	63.10	97.08	136.50	183.21	239.80	307.62	394.66	–
HAS	C1&C2	261.87	276.12	288.90	287.25	306.36	307.68	305.82	301.26	–
VNS	C1	122.05	141.84	218.05	309.12	375.65	502.25	640.32	788.24	–
CLTRS	C1	–	–	–	–	–	–	–	–	54(BR1-BR15)
	C1&C2	–	–	–	–	–	–	–	–	320(BR1-BR15)
FDA	C1	82.94	160.77	298.95	497.79	861.37	1775.79	2218.17	3531.71	–
ID-GLTS	C1	–	–	–	–	–	–	–	–	503.45(BR1-BR15)
	C1&C2	–	–	–	–	–	–	–	–	150.55(BR1-BR15)
HBMLS	C1	537.41	730.33	874.59	1050.7	1161.61	1145.13	1256.03	1255.71	–
	C1&C2	454.76	603.94	722.46	842.52	956.2	1019.06	1129.06	1152.71	–
TS_CLP	C1, C2&C3	202.3	243.7	302.1	346.4	440.3	476.7	530.9	589.5	391.5

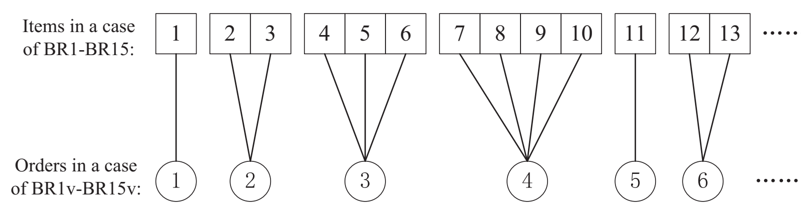

Fig. 3. The relationships between the orders and the items.

the items are not sufficient for the last order, the order will contain insufficient items.

By considering all of the orders as expiring ones, we run **HCLPMC** 20 times for each case in BR1v-BR15v. Next, we select the best result as the upper bound (UB for short) of the corresponding case. We divide the sequence of orders in each case of BR1v-BR15v into two subsets (named sub1 and sub2) according to 9 different ratios that are 9:1, 8:2, . . ., 1:9. If the order number of sub1 is not an integer, it will be rounded up (e.g., 17.6→18). Accordingly, the order number of sub2 will be rounded down. If the order sequence contains q orders, sub1 consists of $or_1, or_2, \ldots, or_j$, whereas sub2 consists of $or_{j+1}, or_{j+2}, \ldots, or_q$. For each proportion, the orders in sub1 are expiring, while the orders in sub2 are non-expiring. Thus, for each one of our 1500 cases, we run **HCLPMC** 18 times. With a total of 27000 running times, **HCLPMC** may be fully tested.

Tables 6–8 list the best, worst and mean results of **HCLPMC** on BR1v-BR15v, respectively. Table 9 lists the mean computational times of **HCLPMC** on BR1v-BR15v.

As shown in Table 6, the best results of BR1v-BR15v range from 85.03% to 89.88%. Table 7 displays that the worst results of BR1v-BR15v range from 80.99% to 86.76%. Table 8 reports that the mean results of BR1v-BR15v range from 83.28% to 88.36%. This proves that **HCLPMC** can produce acceptable results for the container loading problems with the five constraints of C1-C5.

As shown in Table 9, the mean computation times of BR1v-BR15v for all ratios are less than 921 s. This indicates that **HCLPMC** can solve container loading problems with the five constraints (**C1-C5**) in an acceptable time.

To test HCLPMC further, we created 100 new benchmark instances. Each new instance consists of 100 orders and a 20′ feet container (569x213x218 (centimetres) in length, width and height, respectively). Each order consists of 1–6 types of items. The number of item types in each order is generated randomly by using 2016 as the seed. The length, width and height of each item type are integers ranging from 30 to 120. They are generated randomly by using 2016 as the seed. Every time 3 numbers are generated in sequence. The maximum one, the minimum one, and the middle

Table 6
The Best Results of **HCLPMC** on BR1v-BR15v.

	UB (%)	Filling rate (%) under ratio								
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
BR1v	89.11	88.57	88.99	88.83	88.27	88.49	88.51	88.59	88.64	88.88	88.64
BR2v	89.74	89.03	89.36	88.9	89.54	88.81	89.44	89.22	89.14	88.99	89.16
BR3v	90.44	89.7	90.12	90.16	89.57	89.9	89.83	89.77	90.25	89.64	89.88
BR4v	90.04	89.55	89.4	89.09	89.5	89.68	89.45	89.52	89.82	89.35	89.48
BR5v	90.23	89.42	89.23	90.05	89.58	89.75	89.31	90	89.38	89.84	89.62
BR6v	89.72	89.31	89.32	89.46	88.99	89.54	88.98	88.89	89.31	89.21	89.22
BR7v	89.79	89.56	89.41	88.98	88.95	89.24	88.94	89.56	89.47	89.59	89.3
BR8v	89.11	88.72	88.34	88.9	88.8	88.41	88.07	88.48	88.47	88.38	88.51
BR9v	88.85	87.84	88.16	87.59	88.27	87.82	87.83	88.33	88.5	87.66	88
BR10v	88.05	87.39	86.82	87.13	87.58	87.26	86.66	87.6	87.29	86.68	87.16
BR11v	87.93	87.56	87.25	86.91	87.19	87.12	87.44	86.56	86.67	87.32	87.11
BR12v	87.24	86.84	86.57	85.98	86.54	86.6	86.31	86.73	86.96	85.83	86.48
BR13v	86.95	86.14	85.68	86.01	85.48	86.08	86.34	85.56	86.17	86.46	85.99
BR14v	86.84	86.16	86.44	85.24	86.52	85.33	86.54	85.42	86.32	86.41	86.04
BR15v	86.03	85.06	85.21	84.72	85.04	85.28	84.79	85.47	84.7	84.97	85.03

Table 7
The Worst Results of **HCLPMC** on BR1v-BR15v.

	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
BR1v	89.11	87.04	86.45	85.91	86.05	85.89	86.47	86.45	86.82	86.89	86.44
BR2v	89.74	86.97	86.42	85.85	85.55	85.86	85.89	86.25	86.2	86.44	86.16
BR3v	90.44	87.58	86.74	86.75	86.54	86.73	86.32	86.66	86.7	86.81	86.76
BR4v	90.04	87.18	86.63	86.46	86.56	86.8	86.56	86.76	86.68	86.61	86.69
BR5v	90.23	87.41	86.67	86.24	85.72	85.89	85.97	86.53	86.76	86.38	86.4
BR6v	89.72	87.07	86.4	86.34	85.76	85.92	86.36	86.39	86.13	86.04	86.27
BR7v	89.79	86.6	85.53	85.53	85.39	85.55	85.37	85.53	85.31	85.03	85.54
BR8v	89.11	85.43	84.33	84.71	84.23	84.1	84.4	84.47	84.08	84.18	84.44
BR9v	88.85	84.48	83.77	83.63	83.67	83.42	83.61	83.73	83.71	83.27	83.7
BR10v	88.05	84.11	82.84	82.75	83.08	83.14	82.37	83.01	82.62	82.22	82.9
BR11v	87.93	83.02	82.03	82.27	81.8	81.94	82	82.19	81.94	82.01	82.13
BR12v	87.24	83.1	81.41	82.21	81.22	81.42	81.73	82.02	81.31	82.02	81.83
BR13v	86.95	82.67	81.14	81.17	81.05	80.92	81.19	81.74	81.88	81.42	81.46
BR14v	86.84	82.36	80.58	80.37	81.38	81.3	81.01	80.94	81.08	80.5	81.06
BR15v	86.03	82.18	80.36	81.08	80.77	80.6	80.51	81.12	81.1	81.23	80.99

Table 8
The Mean Results of **HCLPMC** on BR1v-BR15v.

	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
BR1v	89.11	87.96	87.44	87.48	87.19	87.3	87.72	87.8	87.75	87.64	87.59
BR2v	89.74	88.18	87.97	87.46	87.53	87.57	87.62	87.76	87.47	87.99	87.73
BR3v	90.44	88.75	88.16	88.33	88.22	88.34	88.19	88.25	88.44	88.52	88.36
BR4v	90.04	88.61	88.02	88	87.97	87.98	88.13	88.45	88.21	88.4	88.2
BR5v	90.23	88.57	88.18	88.11	87.92	88.17	87.89	88.35	88.27	88.03	88.17
BR6v	89.72	88.18	87.85	87.91	87.59	87.56	87.77	88.02	87.47	87.63	87.78
BR7v	89.79	88.04	87.5	87.15	87.48	87.61	87.22	87.44	87.37	87.11	87.44
BR8v	89.11	87.1	86.56	86.6	86.19	86.55	86.46	86.61	86.35	86.54	86.55
BR9v	88.85	86.44	86.08	85.68	85.8	86	85.93	85.54	86.02	85.9	85.93
BR10v	88.05	85.84	85.28	84.92	84.79	85.4	84.8	85.34	84.83	84.72	85.1
BR11v	87.93	85.37	84.83	84.95	84.76	84.85	84.47	84.21	84.75	84.43	84.74
BR12v	87.24	84.91	84.06	84.1	84	84.03	84.57	83.97	84.12	83.89	84.18
BR13v	86.95	84.28	83.88	83.61	83.88	83.63	83.98	83.77	84.14	83.75	83.88
BR14v	86.84	83.95	83.34	83.23	83.75	83.37	83.49	83.53	83.45	83.14	83.47
BR15v	86.03	83.9	83.21	83.09	82.97	83.25	82.94	83.43	83.41	83.29	83.28

Table 9
The Mean computation times of **HCLPMC** on BR1v-BR15v.

	Mean computation time (s) under ratio									
	9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
BR1v	71.6	50.3	50.2	49.5	55.2	49.2	39.1	37.8	41.7	53.1
BR2v	54.3	38.9	36.0	34.7	32.7	30.2	31.6	30.9	29.1	37.9
BR3v	84.6	66.1	55.2	53.2	56.1	52.1	55.1	53.2	55.7	62.2
BR4v	110.3	75.8	73.1	68.7	69.1	62.3	60.4	66.1	61.3	76.1
BR5v	116.6	85.4	80.5	79.3	80.4	68.9	69.3	67.2	69.3	84.3
BR6v	145.7	109.3	98.7	100.1	98.4	84.6	84.1	85.6	86.4	104.7
BR7v	203.9	135.6	134.6	127.3	127.1	113.5	113.7	113.6	116.4	138.7
BR8v	279.8	201.1	195.6	196.1	188.7	162.2	162.2	159.8	162.1	199.4
BR9v	369.5	248.9	237.5	238.3	236.9	205.9	207.5	208.0	209.0	251.8
BR10v	427.6	321.9	305.2	310.0	300.5	265.5	264.1	264.3	271.9	318.4
BR11v	654.3	411.2	374.2	378.5	371.1	316.2	322.1	317.6	319.2	401.8
BR12v	721.1	329.6	298.0	303.8	308.3	368.5	379.1	382.6	386.5	415.1
BR13v	610.3	355.0	216.8	212.0	212.5	423.1	426.1	427.1	435.4	382.5
BR14v	426.8	257.3	239.6	228.8	234.1	480.2	484.2	485.8	492.7	367.5
BR15v	920.7	556.6	551.0	570.1	398.4	541.9	539.1	563.0	556.6	586.8

one are the length, width, and height, respectively. The number of dimensions that can be parallel to the container height when the box is placed in the container ranges from 1 to 3. The number is generated randomly by using 2016 as the seed. When the number is 1, only the box height can be parallel to the container height. When the number is 2, only the height and width of the box can be parallel to the container height. When the number is 3, the height, width and length of the box can all be parallel to the container height. We divided the 100 orders of each instance into expiring ones and non-expiring ones according to nine proportions (which are 9:1, 8:2, 7:3, 6:4, 5:5, 4:6, 3:7, 2:8, and 1:9). For each proportion of each instance, we run HCLPMC 20 times and display the best, worst and mean results, respectively.

Without considering the loading priorities of the orders, we run HCLPMC 20 times to get 20 results for each instance. Next, we select the best result as the upper bound of the results of each

Table 10
The Best Results of **HCLPMC** on our case1-case100.

Case	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
1	85.38	84.92	85.36	84.94	83.25	82.98	83.05	83.03	82.1	83.55	83.69
2	86.07	82.2	82.49	83.83	83.78	85.84	84.91	83.29	79.84	81.85	83.11
3	85.99	83.35	82.35	82.51	85.82	83.5	83.51	83.55	82.15	82.67	83.27
4	86.38	83.25	83.99	85.28	82.94	81.94	86.32	83.88	80.9	82.46	83.44
5	87.26	84.49	83.63	83.92	87.11	85.17	85.89	83.84	83.2	83.98	84.58
6	85.97	84.42	84.47	84.58	84.47	85.62	83.67	83.24	83.96	83.74	84.24
7	84.88	83.12	84.7	83.77	82.26	82.55	83.33	83.81	80.36	82.36	82.92
8	85.35	83.77	83.59	83.13	82.8	83.72	82.73	82.71	84.28	85.29	83.56
9	84.65	83.46	83.99	82.79	84.47	83.62	83.73	84.33	84.38	82.38	83.68
10	86.48	84.4	83.85	83.34	84.17	85.29	83.79	86.42	85.82	82.44	84.39
11	85.44	83.92	83.38	85.31	82.91	82.14	83.41	83.41	84.93	82.45	83.54
12	85.69	83.89	85.58	84.36	82.42	82.53	82.88	84.08	84.01	83.96	83.75
13	84.59	84.1	83.13	84.08	84.47	83.03	82.96	84.51	83.62	81.66	83.51
14	84.87	83.46	84.29	83.6	82.65	82.42	84.11	84.55	82.86	82.55	83.39
15	85.39	82.06	82.79	83.25	83.12	83.75	82.26	85.2	83.26	82.27	83.11
16	86.19	85.61	84.34	85.06	83.96	83.29	84.62	85.34	86.16	85.4	84.86
17	86.89	83.17	84.28	84.22	84.22	84.16	82.98	86.81	84.88	84.25	84.33
18	86.06	82.96	83.26	85.85	82.99	81.99	83.85	85.44	83.46	82.46	83.58
19	85.59	84.71	81.82	84.05	84.66	83.65	85.3	83.59	84.38	82.41	83.84
20	84.82	84.09	81.9	82.83	84.76	84.32	82.76	81.93	84.73	82.95	83.36
21	85.62	83.99	83.52	83.68	82.76	84.74	85.48	83.3	83.98	81.01	83.61
22	86.1	84.23	83.2	84.17	84.75	86.04	85.73	82.91	84.75	81.75	84.17
23	85.47	85.15	85.18	83.23	84	83.61	85.07	83.78	82.79	82.55	83.93
24	85.34	83.71	83.31	84.93	85.15	84.28	84.7	83.21	83.4	80.32	83.67
25	86.52	85.6	85.31	86.14	85	83.55	84.63	82.22	81.6	82.85	84.1
26	86.49	83.89	82.91	86.25	84.38	83.29	81.07	83.38	81.35	82.42	83.22
27	86.93	84.28	83.38	83.32	84.24	84.78	86.84	84.15	84.06	83.89	84.33
28	84.94	83.59	84.1	83.7	83.64	82.16	84.77	84.46	81.78	81.22	83.27
29	86.79	84.18	86.45	82.82	82.44	84.09	84.85	82.8	83.39	82.76	83.75
30	87.03	86.84	85.51	85.05	84.35	84.38	84.4	83.99	83.03	82.08	84.4
31	85.05	84.25	81.72	84.99	84.02	83.5	83.09	81.94	83.14	84.14	83.42
32	85.54	85.2	84.27	83.32	83.7	82.58	83.32	83.72	81.78	82.78	83.41
33	85.75	85.38	81.7	83.39	84.13	85.14	84	82.26	82.95	80.26	83.25
34	85.39	83.14	83.4	85.37	84.02	81.95	83.51	83.45	83.36	83.52	83.52
35	86.49	83.89	82.7	84.64	83.66	86.13	82.62	85.45	85.27	83.3	84.18
36	85.1	84.72	82.5	83.1	84.29	82.61	83.61	84.25	82.53	82.3	83.32
37	84.53	82.68	82.95	84.5	83.41	82.52	84.17	82.93	82.84	82.46	83.16
38	85.7	83.45	83.64	85.49	83.42	83.96	84.13	85.54	83.73	83.68	84.12
39	86.19	83.97	81.72	85.01	86.14	83.54	84.67	84.05	85.14	84.15	84.27
40	85.26	83.33	84.1	82.92	84.08	83.07	85.02	82.13	82.67	81.67	83.22
41	84.51	83.49	82.19	83.1	83.82	84.19	84.38	84.46	82.85	81.5	83.33
42	85.59	85.23	82.92	85.38	84.52	84.63	85.14	84.41	81.42	83.42	84.12
43	87.33	83.92	84.76	85.18	87.09	83.04	83.1	82.67	84.02	82.68	84.05
44	84.31	83.52	83.07	83.08	81.86	82.59	83.94	83.52	83.75	82.59	83.1
45	85.12	83.95	82.98	85.1	82.98	85.06	84.09	83.69	82.46	81.73	83.56
46	84.78	83.59	83.62	83.47	84.66	82.8	83.51	83.31	83.94	82.94	83.54
47	86.93	86.64	84.94	85.14	83.45	83.72	82.6	83.5	84.44	84.38	84.31
48	86.95	86.87	82.22	82.93	83.13	85.76	83.37	85.43	84.03	82.44	84.02
49	87.07	84.98	85.05	83.22	83.7	86.09	84.83	86.74	83.18	83.78	84.62
50	87.43	83.67	82.58	86.39	84.42	83.09	87.19	84.26	84.32	81.42	84.15
51	85.58	83.76	82.8	82.8	85.55	83.72	83.09	85.27	83.68	83.28	83.77
52	85.98	85.02	83.76	84.57	84.2	84.79	85.81	84.69	83.89	82.9	84.4
53	86.89	86.67	83.63	82.9	84.77	84.32	83.85	84.39	85.38	84.4	84.48
54	84.1	83.29	83.78	83.77	83.91	81.26	82.78	82.82	82.78	81.79	82.91
55	86.56	85.53	83.71	86.45	83.68	85.24	84.53	86.24	83.44	82.3	84.57
56	86.04	82.44	84.73	85.69	83.81	85.23	83.68	85.08	80.81	82.81	83.81
57	85.32	84.99	82.3	83.79	83.2	83.7	83.74	83.36	81.83	81.83	83.19
58	84.78	83.68	83.63	83.44	84.41	82.28	82.12	83.16	84.02	84.02	83.42
59	86.26	85.67	82.44	82.43	84.3	86.19	85.06	84.71	83.5	81.86	84.02
60	86.79	85.06	83.62	85.8	86.49	85	85.56	83.77	82.82	79.82	84.22
61	85.37	83.16	84	84.19	82.45	85.11	84.54	84.97	84.77	81.87	83.9
62	85.24	84.51	82.84	82.83	82.83	83.79	84.56	83.14	85.23	84.33	83.78
63	85.3	82.32	82.02	85.19	82.72	83.25	83.31	84.29	84.92	81.93	83.33
64	87.76	84.52	84.09	87.39	83.72	82.97	84.56	84.32	84.06	82.37	84.22
65	86.19	83.7	83.69	84.01	86.19	82.9	84.55	82.65	84.18	85.19	84.12
66	86.96	85.21	83.87	82.65	85.71	83.51	83.87	86.79	84.14	81.8	84.17
67	84.47	84.1	84.15	82.48	84.34	81.93	84.35	84.08	83.96	81.51	83.43
68	83.83	83.78	81.83	83.1	82.23	83.72	83.17	82.3	82.31	81.3	82.64
69	84.73	82.69	83.43	84.39	83.81	83.81	83.97	84.16	82	80.84	83.23
70	85.44	85.03	83.77	83.76	82.27	83.85	82.52	85.25	83.95	80.47	83.43
71	87.03	85.5	85.09	84.58	84.03	83.89	84.49	85.61	86.65	83.66	84.83
72	87.36	86.24	83.72	87.06	85.02	83.2	84.3	83.67	84.67	83.74	84.62
73	85.66	82.84	83.75	85.5	82.79	84.54	83.57	83.78	82.91	82.1	83.53

(*continued on next page*)

Table 10 (*continued*)

Case	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
74	84.32	84.08	83.12	83.41	83.71	84.28	82.36	84.19	83.06	80.88	83.23
75	84.89	84.31	82.62	84.7	82.42	83.59	84.67	84.15	83.28	84.88	83.85
76	84.63	84	81.93	83.41	82.7	84.06	84.01	83.56	84.32	82.77	83.42
77	85.54	85.33	83.98	83.98	83.71	85.14	84.52	83.78	83.47	82.31	84.02
78	85.45	85.15	83.84	84.89	83.91	82.5	82.66	81.95	83.36	82.69	83.44
79	84.62	83.36	84.38	83.26	84.18	83.41	83.75	83.69	83.8	83.81	83.74
80	86.35	83.18	84	83.24	83.79	85.23	86.1	85.27	83.31	83.31	84.16
81	86.22	84.87	84.75	83.87	83.17	85.88	82.93	82.2	84.67	80.42	83.64
82	85.03	82.53	82.85	82.68	85.02	83.06	84.6	84.12	84.31	80.14	83.26
83	87.18	82.88	82.15	83.65	86.96	83.64	83.4	82.66	81.65	81.67	83.18
84	87.07	84.83	86.97	82.62	82.57	83.59	84.53	84.05	84.06	82.06	83.92
85	85.94	83.38	81.82	82.68	85.07	85.79	83.38	83.07	84.14	82.14	83.5
86	84.58	83.1	84.06	82.15	84.31	83.24	82.58	81.63	82.53	83.53	83.01
87	86.31	86.2	85.41	83.9	85.01	82.42	84.4	82.34	83.86	81.47	83.89
88	85.88	84.02	83.08	83.59	83.55	84.58	85.35	83.05	85.7	81.61	83.84
89	84.96	82.5	84.23	84.07	84.66	83.58	84.31	83.55	83.72	81.67	83.59
90	85.33	82.8	81.96	85.32	83.25	83.27	84.8	84.72	84.27	82.27	83.63
91	84.89	83.41	82.57	84.85	83.61	83.84	84.09	82.82	82.51	82.21	83.32
92	84.86	84.62	83.97	84.46	84.31	84.24	83.96	84.81	82.41	81.41	83.8
93	86.23	83.68	83.3	85.66	82.1	84.75	86.14	83.63	82.55	82.97	83.86
94	85.34	84.33	83.69	83.31	84.3	85.15	83.81	85.2	83.9	83.91	84.18
95	86.76	86.39	82.77	85.65	84.76	85.16	85.91	83.94	81.66	84.24	84.5
96	84.75	84.39	83.49	83.13	82.47	84.08	83.71	83.66	84.71	81.71	83.48
97	84.99	82.14	83.19	83.92	83.4	83.97	84.4	82.39	82.71	81.47	83.07
98	84.62	83.08	82.14	84.07	82.95	84.37	83.55	83.6	83.79	82.64	83.35
99	84.96	84.06	83.31	84	83.3	83.48	83.36	83.55	83.75	84.59	83.71
100	85.28	83.22	83.24	84.76	82.6	81.78	82.19	83.77	85.05	80.79	83.04

instance. Tables 10–12 list the best, worst and mean results of **HCLPMC** on the 100 instances, respectively. Table 13 lists the mean computation times of **HCLPMC** on the 100 instances.

As shown in Table 10, the best results of the 100 instances range from 82.64% to 84.86%. Table 11 displays that the worst results of the 100 instances range from 78.7% to 81.63%. Table 12 reports that the mean results of the 100 instances range from 80.87% to 83.31%. This proves that **HCLPMC** can produce acceptable results for the container loading problems with the five constraints of C1-C5.

As shown in Table 9, the mean computation times of the 100 iterations for all ratios are less than 1197 s. This indicates that **HCLPMC** can solve container loading problems with the five constraints (**C1**-**C5**) in an acceptable time.

Table 11
The Worst Results of **HCLPMC** on our case1-case100.

Case	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
1	85.38	80.77	80.14	80.35	79.36	79.16	80.58	80.1	79.66	78.55	79.85
2	86.07	79.15	81.35	80.17	81.83	80.83	80.17	79.04	77.83	76.85	79.69
3	85.99	78.06	80.23	80.16	80.17	80.59	79.84	78.9	78.9	76.67	79.28
4	86.38	77.69	81.52	81.1	80.99	78.64	78.44	78.59	79.37	78.9	79.47
5	87.26	81.46	79.52	81.4	82.66	80.64	81.91	79.89	81.45	79.98	80.99
6	85.97	79.8	80.94	80.68	81.28	81.56	80.58	81.89	80.27	81.74	80.97
7	84.88	78.32	79.98	79.64	78.89	79.47	78.92	78.98	75.74	78.36	78.7
8	85.35	81.11	80.88	80.88	79.82	80.05	79.31	80.81	81.4	80.29	80.51
9	84.65	80.87	81.79	80.82	81.73	80.72	82.66	81.82	81.69	79.27	81.26
10	86.48	80.77	81.01	81	81.34	81.4	80.44	81.34	81.45	80.44	81.02
11	85.44	80.32	79.72	80.68	80.61	79.63	80.4	80.93	78.98	79.45	80.08
12	85.69	80.61	79.65	79.63	80.55	80.94	80.41	79.92	80.61	79.12	80.16
13	84.59	80.96	80.09	83.1	81.14	80.94	80.47	81.1	78.57	78.66	80.56
14	84.87	80.11	80.73	80.73	81.41	79.56	79.72	81.53	82.08	80.11	80.66
15	85.39	80.57	78.94	80.56	80.19	79.79	80.71	81.75	79.85	80.27	80.29
16	86.19	78.89	78.89	80.12	81.81	81.39	81.28	80.68	80.46	80.4	80.44
17	86.89	80.41	80.13	81.36	81.28	79.09	80.66	80.83	80.25	79.88	80.43
18	86.06	81.03	78.93	79.87	80.32	81.3	79.03	82.41	81.25	79.46	80.4
19	85.59	80.85	79.77	82.02	79.53	81.48	81.49	82.68	81.11	78.41	80.82
20	84.82	80.69	78.68	80.58	81.53	80.06	80.54	79.63	80.93	79.73	80.26
21	85.62	80.65	80.65	79.66	80.98	80.67	80.44	80.61	80.62	79.01	80.37
22	86.1	79.97	81.27	80.25	80.68	80.66	78.35	81.7	82.66	79.66	80.58
23	85.47	79.17	80.14	82.51	80.31	77.97	80.53	79.23	77.55	79.79	79.69
24	85.34	81.5	79.82	79.94	79.79	79.83	78.97	82.23	79.16	78.32	79.95
25	86.52	81.62	81.47	82.72	81.45	80.8	81.43	79.46	80.69	79.23	80.99
26	86.49	81.34	79.88	80.55	80.92	80.03	79.41	80.34	79.27	78.69	80.05
27	86.93	79.18	81.38	80.42	80.03	81.25	81.18	80.29	82.23	78.89	80.54

Table 11 (*continued*)

Case	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
28	84.94	80.83	80.83	81.74	81.19	79.84	79.61	79.56	79.66	78.22	80.16
29	86.79	82.45	80.31	81.11	80.22	82.76	80.72	80.47	80.96	81.56	81.17
30	87.03	78.6	79.57	81.48	81.29	80.56	79.97	81.37	79.19	80.08	80.23
31	85.05	81.67	79.75	83.05	81.56	82.6	80.92	79.7	81.14	80.14	81.17
\32	85.54	81.4	79.5	80.63	80.71	78.77	81.39	80.4	79.76	79.76	80.26
33	85.75	79.42	79.75	80.71	79.57	79.58	80.69	80.59	79.75	77.26	79.7
34	85.39	81.18	80.6	82.51	82.25	80.29	79.49	80.84	81.52	81.52	81.13
35	86.49	81.46	79.78	82.48	81.12	83.58	81.61	83.16	81.18	80.3	81.63
36	85.1	79.61	78.03	80.74	80.91	80.42	80.26	79.32	79.76	80.3	79.93
37	84.53	80.06	80.83	80.05	79.84	79.82	80.78	80.95	80.23	80.23	80.31
38	85.7	81.21	80.39	80.37	80.18	81.44	81.19	81.06	81.67	80.68	80.91
39	86.19	82.42	80.65	82.57	79.47	82.32	81.43	81.47	81.72	81.15	81.47
40	85.26	80.23	79.89	80.39	80.2	81.17	79.32	79.13	80.65	78.65	79.96
41	84.51	80.43	80.09	79.7	81.04	80.1	79.11	79.94	80.21	78.5	79.9
42	85.59	80.99	80.89	81.25	78.95	81.71	80.97	81.13	78.75	79.42	80.45
43	87.33	80.25	81.21	82.18	81.19	81.29	80.11	80.7	80.08	79.7	80.75
44	84.31	80.34	79.21	79.2	80.15	79.38	78.99	80.16	81.63	77.65	79.63
45	85.12	81.68	80.58	80.28	80.28	80.51	81.59	79.97	81.04	79.73	80.63
46	84.78	78.07	81.51	82.3	80.84	79.59	80.38	79.57	79.37	78.38	80
47	86.93	81.28	80.82	80.99	81.11	79.65	80.64	80.8	80.22	80.22	80.64
48	86.95	81.28	80.05	80.41	80.55	80.96	80.38	81.6	79.63	78.63	80.39
49	87.07	79.36	81.02	78.93	81.2	79.36	83.1	81.27	79.89	80.78	80.55
50	87.43	79.07	80.77	82.18	80.3	81.2	81.33	81.19	83.11	79.33	80.94
51	85.58	77.99	80.87	80.91	80.83	80.12	79.61	82.25	79.26	78.1	79.99
52	85.98	81.39	80.42	82.33	81.33	82.35	81.47	80.51	80.73	78.9	81.05
53	86.89	79.05	81.94	79.97	80.92	81.86	81.9	81.61	81.62	79.42	80.92
54	84.1	80.47	81.25	80.26	80.28	79.85	81.24	80.22	80.26	79.38	80.36
55	86.56	78.81	79.78	80.26	80.24	82.23	80.5	82.41	82.02	81.3	80.84
56	86.04	81.44	79.61	80.48	80.22	82.22	82.47	81.41	78.81	79.81	80.72
57	85.32	79.4	80.36	80.8	80.19	79.17	79.38	80.13	79.52	79.83	79.86
58	84.78	78.83	81.37	80.81	79.78	79.41	79.05	81.04	81.45	82.13	80.43
59	86.26	78.43	80.25	82.13	79.77	81.66	81.55	81.19	80.73	78.69	80.49
60	86.79	80.95	79.6	80.91	82.35	80.18	81.25	80.81	80.8	77.82	80.52
61	85.37	80.48	81.13	81.35	81.17	82.05	82.22	81.79	83.12	79.58	81.43
62	85.24	79.94	80.83	79.82	80.9	81.79	81.77	79.92	81.31	81.32	80.84
63	85.3	79.48	79.48	81.02	79.45	79.35	80.96	79.36	80.11	79.93	79.9
64	87.76	80.88	81.83	81.77	80.3	81.26	80.73	82.28	81.52	79.37	81.1
65	86.19	78.88	79.84	80.23	80.79	80.56	81.2	79.66	80.67	77.67	79.94
66	86.96	80.35	82.08	79.75	80.52	79.31	81.17	82.68	81.54	79.8	80.8
67	84.47	82.17	79.77	79.75	79.96	80.99	79.86	80.01	81.44	79.51	80.38
68	83.83	80.42	78	80.89	79.36	80.87	79.16	79.91	79.7	78.3	79.62
69	84.73	80.52	80.64	80.79	79.52	79.5	79.81	81.07	81.83	76.01	79.97
70	85.44	81.64	80.07	81.42	79.6	81.93	80.61	80.92	80.47	79.05	80.63
71	87.03	81.49	81.19	82.44	82.72	81.11	80.21	82.12	83.34	80.01	81.63
72	87.36	80.3	81.83	82.22	81.01	81.24	81.66	81.09	78.47	77.2	80.56
73	85.66	79.76	80.45	80.68	80.26	81.6	80.7	81.67	78.78	80.1	80.44
74	84.32	79.26	79.58	79.58	79.39	81.24	80.31	81.21	80.37	78.63	79.95
75	84.89	81.38	80.73	80.4	81.65	81.74	80.65	82.3	80.1	80.11	81.01
76	84.63	80.45	79.54	81.29	80.12	81.16	80.61	79.8	81.27	79.18	80.38
77	85.54	80.12	81.09	80.89	81.39	82.01	81.07	81.98	81.43	80.53	81.17
78	85.45	82.14	79.88	81.06	80.71	79.95	77.81	80.07	81.32	81.51	80.49
79	84.62	80.41	78.57	80.62	79.28	81.44	79.46	80.79	78.96	78.81	79.82
80	86.35	81.69	81.69	80.52	80.83	80.84	81.83	84.12	80.3	79.41	81.25
81	86.22	81.98	80.07	80.99	79.99	81.02	79.34	78.42	79.3	77.42	79.84
82	85.03	78.71	79.8	80.83	81.54	80.53	80.6	81.08	81.02	79.15	80.36
83	87.18	79.38	79.27	80.44	80.61	81.13	80.24	80.2	79.59	77.67	79.84
84	87.07	79.55	80.69	81.04	80.64	80.57	79.7	82.5	79.66	79.06	80.38
85	85.94	79.64	79	80.37	80.65	80.67	79.75	79.57	79.79	78.14	79.73
86	84.58	81.2	80.23	80.59	79.24	79.19	79.51	79.45	79.52	79.53	79.83
87	86.31	81.8	78.91	80.61	80.96	81.82	78.95	79.77	82.02	79.23	80.45
88	85.88	80.9	82.11	81.13	81.31	81.47	82.06	80.58	78.08	79.7	80.82
89	84.96	79.51	79.27	80.46	80.06	79.99	80.1	80.33	80.79	78.67	79.91
90	85.33	80.65	79.91	81.7	81.6	80.78	81.76	80.61	80.95	79.27	80.8
91	84.89	79.48	79.92	80.56	80.44	80.44	79.07	79.54	78.21	77.9	79.51
92	84.86	83.58	81.07	79.13	82.06	81.93	81.24	81.99	81.52	80.51	81.45
93	86.23	80.03	79.51	79.86	78.89	81.39	80.18	81.4	80.69	79.97	80.21
94	85.34	81.68	82.37	81.67	81.33	81.45	80.71	82.23	81.9	79.9	81.47
95	86.76	80.53	80.97	81.41	81.82	80.7	79.78	81.08	80.5	80.97	80.86
96	84.75	79.64	79.3	80.25	80.14	80.68	81.68	79.42	79.57	76.95	79.74
97	84.99	79.1	79.39	80.35	79.79	79.31	80.95	80.06	76.33	76.33	79.07
98	84.62	80.06	79.24	80.18	81.01	80.07	78.28	80.05	78.49	77.45	79.43
99	84.96	80.36	82.1	81.79	80.99	80.43	80.2	82.33	82.36	81.59	81.35
100	85.28	80.32	81.35	80.32	79.36	79.26	80.12	80.71	80.18	77.2	79.87

Table 12
The Mean Results of **HCLPMC** on our case1-case100.

Case	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
1	85.38	82.69	82.36	82.77	82.02	80.94	82	81.66	81.3	80.77	81.83
2	86.07	81.06	81.7	81.93	82.68	83.3	82.34	80.75	79.17	79.37	81.37
3	85.99	80.75	81.38	81.67	81.91	82.13	81.47	80.39	80.68	79.73	81.12
4	86.38	80.65	82.61	82.71	82.09	81.01	82.38	82.02	79.8	80.75	81.56
5	87.26	82.71	82.3	83.08	84.23	82.74	83.61	81.91	82.77	82.39	82.86
6	85.97	82.08	82.57	82.69	83.06	83.03	82.26	82.72	82.81	82.78	82.67
7	84.88	80.48	81.61	81.36	81.22	81.53	81.05	81.92	79	79.64	80.87
8	85.35	82.52	82.34	82.54	81.74	81.67	81.19	81.85	83.16	82.55	82.17
9	84.65	82.35	82.62	81.8	83	82.14	82.93	83.23	82.93	81.15	82.46
10	86.48	82.61	82.6	81.93	82.22	82.8	81.91	83.14	83.96	81.51	82.52
11	85.44	81.81	81.81	82.58	81.96	81.26	81.65	82.62	83.07	80.57	81.93
12	85.69	82.15	82.01	81.85	81.28	81.55	81.79	81.71	82.5	80.99	81.76
13	84.59	82.07	81.71	83.59	82.97	82.06	81.83	82.69	80.75	80.03	81.97
14	84.87	81.33	81.88	82.12	82.02	81	81.88	83.14	82.56	81.6	81.95
15	85.39	81.24	80.95	81.9	81.89	82.26	81.43	82.92	82.24	81.2	81.78
16	86.19	81.96	80.81	82.01	83.02	82.23	82.99	82.9	83.64	82.63	82.47
17	86.89	81.68	82.06	82.95	82.6	80.75	81.98	83.14	82.5	81.55	82.13
18	86.06	82.18	81.26	82.71	81.5	81.61	81.94	84.22	82.1	81.42	82.1
19	85.59	82.33	80.58	82.93	82.46	83.04	83.59	83.11	83.14	80.75	82.44
20	84.82	82.38	80.32	81.69	82.5	81.45	81.51	81	82.62	81.11	81.62
21	85.62	83.04	82.27	81.45	81.86	82.69	82.43	82.04	81.37	80.01	81.91
22	86.1	81.4	82.24	82.36	83.08	82.65	82.23	82.41	83.23	80.14	82.19
23	85.47	81.86	82.63	82.84	81.79	81.95	82.27	82.27	80.01	80.74	81.82
24	85.34	82.28	81.32	82.08	81.72	81.51	81.94	82.91	81.24	79.49	81.61
25	86.52	83.35	82.81	84.14	82.82	82.43	83.14	81.21	81.17	80.72	82.42
26	86.49	82.45	81.85	82.61	82.33	81.8	80.66	81.31	80.22	80.68	81.55
27	86.93	81.72	82.63	82.11	82.29	83.23	82.91	82.48	83.26	81.52	82.46
28	84.94	81.78	82.28	82.49	82.45	81.32	81.89	82.19	80.83	80.07	81.7
29	86.79	83.55	82.79	82.18	81.33	83.34	82.76	81.63	82.53	81.92	82.45
30	87.03	81.67	82.91	83.24	82.24	82.63	82.16	82.28	81.97	81.27	82.26
31	85.05	83.18	80.9	83.71	82.55	83.02	82.41	80.59	81.73	81.93	82.22
\32	85.54	83.38	82.26	82.23	82.18	80.97	82.12	81.69	80.54	80.58	81.77
33	85.75	82.04	80.75	82.17	80.94	82.5	82.37	81.43	81.97	78.99	81.46
34	85.39	82.22	82.07	83.51	83.13	81.38	81.56	82.26	82.17	82.21	82.28
35	86.49	83.02	81.04	83.39	82.58	84.67	82.11	84.05	83.07	82.03	82.88
36	85.1	81.16	81.16	82.06	82.42	81.55	81.97	82.06	81.43	80.9	81.63
37	84.53	81.2	81.99	82.59	81.31	81.02	82.5	82.16	81.92	81.21	81.77
38	85.7	82.38	81.78	82.61	81.72	82.94	82.08	82.58	82.68	82.08	82.32
39	86.19	82.93	81.17	83.6	82.28	83.08	83.1	82.36	83.81	82.3	82.74
40	85.26	81.62	81.3	81.94	82.32	82.04	82.4	80.56	82.05	80.86	81.68
41	84.51	81.68	81.23	81.16	82.3	82.04	81.64	82.73	81.71	80.04	81.61
42	85.59	82.06	82.09	82.46	81.98	82.72	83.09	82.26	79.58	80.97	81.91
43	87.33	82.4	82.77	83.63	83.4	82.31	81.65	82.16	82.1	81.29	82.41
44	84.31	81.65	81.36	80.92	80.89	80.55	80.94	81.68	82.73	81.2	81.32
45	85.12	82.88	81.81	82.95	81.67	82.95	82.94	81.98	81.65	80.8	82.18
46	84.78	81.18	82.68	82.95	82.25	81.32	82.25	81.45	81.81	80.23	81.79
47	86.93	84.42	82.61	82.46	82.08	81.69	81.82	82.04	82.36	82.54	82.45
48	86.95	83.4	80.8	81.5	82.15	82.33	82.16	83.44	82.17	80.68	82.07
49	87.07	81.86	82.73	81.73	82.12	82.76	84.07	83.07	81.63	81.81	82.42
50	87.43	81.71	81.6	83.86	82.16	81.9	83.74	82.79	83.75	80.75	82.47
51	85.58	81.42	81.69	82.25	82.82	82.29	81.33	83.34	81.54	80.44	81.9
52	85.98	83.01	81.66	83.19	82.66	83.6	83.1	83.16	82.79	80.44	82.62
53	86.89	82.81	82.72	81.35	83.04	82.79	82.97	82.81	83.9	81.8	82.69
54	84.1	81.55	82.26	82.18	82.13	80.54	81.9	81.39	81.41	80.51	81.54
55	86.56	82.49	82.35	82.5	82.24	83.09	82.79	83.79	82.66	82.08	82.67
56	86.04	81.92	82.3	82.68	81.59	83.24	83.28	83.24	80.02	81.42	82.19
57	85.32	81.8	81.22	82.04	81.79	81.52	82.16	81.49	81.18	80.85	81.56
58	84.78	81.18	82.57	82.07	81.51	80.93	81.04	82.31	82.91	83.04	81.95
59	86.26	81.76	81.43	82.22	82.22	83.04	83.38	82.76	82.24	80.72	82.2
60	86.79	82.85	81.97	82.87	83.76	83.43	83.33	82.08	81.92	79.32	82.39
61	85.37	81.68	82.62	82.61	81.99	83.72	83.07	83.39	83.91	81.04	82.67
62	85.24	81.88	81.47	81.34	82.27	82.39	82.88	81.58	83.88	82.52	82.25
63	85.3	80.61	81.03	83.09	81.26	81.34	82.31	81.93	81.5	80.83	81.54
64	87.76	82.43	83.16	83.26	82.71	82.21	82.97	82.94	82.37	81.19	82.58
65	86.19	81.38	80.98	81.97	82.81	82.27	82.7	81.45	82.12	80.89	81.84
66	86.96	83.47	83.12	81.42	83.67	82.17	82.59	83.99	82.91	81.15	82.72
67	84.47	83.27	82.48	80.83	82.07	81.36	82.4	81.48	82.63	80.63	81.91
68	83.83	81.95	80.36	82.23	81.45	82.33	81.83	81.24	81.1	79.39	81.32
69	84.73	81.19	81.88	82.7	82.1	81.26	81.85	82.12	81.96	78.37	81.49
70	85.44	83.22	81.37	82.4	81.3	82.91	81.69	83.26	81.7	79.73	81.95
71	87.03	83.2	83.28	83.76	83.18	82.46	82.88	83.68	85.05	82.33	83.31
72	87.36	82.99	82.78	84.26	82.65	82.36	82.98	81.92	81.84	81.23	82.56
73	85.66	81.47	82.06	82.23	81.18	82.92	82.02	82.59	81.32	81.23	81.89

Table 12 (*continued*)

Case	UB (%)	Filling rate (%) under ratio									
		9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
74	84.32	80.92	81.58	81.18	81.63	82.25	81.13	82.56	81.76	79.43	81.38
75	84.89	82.66	81.88	82.52	81.95	82.58	82.3	82.93	82.2	81.93	82.33
76	84.63	82.35	80.81	82.52	81.19	82.18	82.75	82.1	82.52	80.46	81.88
77	85.54	82.28	82.88	82.63	82.4	83.27	83.33	83.08	82.42	81.09	82.6
78	85.45	83.96	81.48	82.71	81.98	81.59	80.84	81.22	82.58	81.95	82.03
79	84.62	81.65	81.84	81.89	82.34	82.11	82.32	81.9	81.49	80.85	81.82
80	86.35	82.44	82.89	81.98	82.69	82.75	83.36	84.62	81.92	81.33	82.66
81	86.22	83.15	82.04	82.18	82.21	83.22	81.87	80.94	81.4	79.27	81.81
82	85.03	80.69	80.76	82.08	82.42	82.11	82.92	82.89	82.33	79.43	81.74
83	87.18	81.08	81.02	82.34	82.77	82.37	81.64	81.76	80.48	79.31	81.42
84	87.07	81.86	82.62	82.07	81.69	82.05	82.15	83.02	81.91	80.58	81.99
85	85.94	81.39	80.22	81.47	82.16	82.37	81.52	81.61	82.58	80.05	81.49
86	84.58	81.91	82.2	81.48	81.28	81.41	81.36	80.97	81.35	80.95	81.43
87	86.31	84.06	83.11	81.72	82.72	82.16	81.42	81.3	82.98	80.42	82.21
88	85.88	82.31	82.46	82.24	81.92	82.54	83.46	82.27	81.61	81.03	82.2
89	84.96	81.04	82.2	82.03	81.76	82.04	82.05	81.77	82.27	80.09	81.69
90	85.33	81.81	81.14	83.4	82.72	82.26	82.64	82.53	82.75	80.82	82.23
91	84.89	81.97	81.58	83.17	81.65	82.06	81.28	81.18	80.84	80.57	81.59
92	84.86	84.03	82.21	82.11	83.09	83.11	82.9	83.23	81.69	80.69	82.56
93	86.23	82.1	81.4	82.17	80.76	82.86	82.59	82.38	81.71	81.46	81.94
94	85.34	83.35	83.13	82.49	82.93	83.03	81.82	83.19	82.34	81.94	82.69
95	86.76	83.24	81.85	83.2	83.09	83.06	82.2	82.42	80.96	81.79	82.42
96	84.75	82.36	81.66	82	81.52	82.4	82.78	81.81	82.13	79.76	81.82
97	84.99	80.68	81.5	82.05	81.41	81.26	82.68	81.55	80.82	78.64	81.18
98	84.62	81.69	80.72	81.85	81.84	82.29	80.97	81.84	82.01	79.4	81.4
99	84.96	82.3	82.81	83.04	81.83	81.74	82.36	83.05	83.35	82.74	82.58
100	85.28	82.24	82.16	82.27	80.95	80.61	81.4	81.87	82.55	78.68	81.41

Table 13
The mean computation times of **HCLPMC** on our case1-case100.

Case	Mean computation time (s) under ratio									
	9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
1	808	792	763	774	743	817	798	758	632	765
2	628	703	637	667	872	763	953	565	683	719
3	682	597	494	627	507	445	587	617	503	562
4	708	637	602	722	801	834	627	529	597	673
5	654	743	642	636	740	531	818	873	625	696
6	734	728	535	663	781	782	623	607	423	653
7	879	856	728	793	618	702	900	752	715	771
8	701	751	741	584	845	832	630	700	538	702
9	759	712	684	819	845	787	826	629	597	740
10	673	687	761	603	603	806	649	580	612	664
11	717	670	563	722	856	608	658	534	558	654
12	696	515	713	717	577	713	671	573	472	627
13	876	665	875	738	811	513	736	742	545	722
14	922	960	820	931	826	933	928	868	636	869
15	856	912	801	656	815	630	618	738	663	743
16	755	912	692	695	720	576	622	594	481	672
17	932	668	803	622	722	963	696	536	550	721
18	820	702	724	767	672	836	708	687	636	728
19	745	819	852	721	859	735	814	696	735	775
20	508	601	668	602	744	821	699	540	487	630
21	783	756	679	686	753	497	709	473	578	657
22	907	753	769	727	751	778	796	670	609	751
23	900	697	585	767	735	514	753	569	657	686
24	685	859	965	779	786	861	856	735	552	786
25	650	577	550	598	719	715	695	545	559	623
26	729	709	630	656	656	733	817	659	594	687
27	745	762	541	645	464	528	636	479	395	577
28	893	735	828	690	919	890	828	634	700	791
29	532	557	583	611	586	418	494	660	443	543
30	696	667	644	613	586	651	621	474	524	608
31	460	524	466	528	562	548	690	551	521	539
\32	577	894	731	839	907	901	1054	704	584	799
33	1027	887	920	846	716	890	933	936	769	880
34	877	873	733	905	887	823	768	647	712	803
35	709	583	610	468	398	552	532	466	374	521
36	748	793	805	832	808	729	681	633	644	741
37	574	624	575	576	524	606	697	594	381	572

(*continued on next page*)

Table 13 (*continued*)

Case	Mean computation time (s) under ratio									
	9:1	8:2	7:3	6:4	5:5	4:6	3:7	2:8	1:9	Mean
38	1065	940	923	868	938	799	791	845	585	862
39	716	625	553	511	565	543	629	424	566	570
40	685	817	769	746	917	699	639	465	526	696
41	623	670	734	595	654	647	689	462	611	632
42	806	776	682	717	730	805	661	634	542	706
43	736	724	718	550	612	535	622	514	586	622
44	956	919	901	1058	830	896	769	714	568	846
45	789	700	869	878	796	855	803	682	726	789
46	690	664	525	621	579	576	793	709	535	632
47	610	859	877	843	894	932	1000	809	639	829
48	797	931	828	685	729	712	750	761	634	759
49	863	693	697	746	817	807	731	931	780	785
50	788	849	789	1003	763	687	701	658	659	766
51	777	666	800	670	760	753	574	370	462	648
52	662	646	614	633	501	594	673	652	610	621
53	803	709	738	635	546	638	500	454	427	606
54	812	840	650	553	698	690	708	718	582	695
55	657	714	816	765	657	689	526	479	538	649
56	775	628	569	622	562	605	625	554	591	615
57	747	723	787	804	740	815	657	584	512	708
58	697	684	562	674	630	637	669	921	490	663
59	829	684	875	609	696	793	593	632	642	706
60	724	607	579	689	676	498	706	714	421	624
61	806	824	766	859	756	727	686	408	489	702
62	815	759	696	746	725	704	635	588	578	694
63	823	875	758	799	685	675	759	576	698	739
64	750	828	913	783	1000	977	765	857	820	855
65	661	602	487	654	465	590	509	517	429	546
66	624	762	842	655	703	728	565	664	515	673
67	765	674	676	667	527	641	669	444	596	629
68	831	919	762	683	766	822	647	843	643	768
69	782	772	587	783	756	782	842	643	777	747
70	624	580	557	581	701	567	562	591	380	571
71	567	640	823	731	605	548	577	459	515	607
72	607	697	546	551	632	776	725	676	506	635
73	781	992	833	694	675	898	623	607	659	751
74	1197	1004	985	945	945	882	838	838	736	930
75	785	691	664	741	600	693	661	746	550	681
76	800	776	745	681	678	645	706	540	575	683
77	687	507	611	598	508	701	482	442	569	567
78	525	840	520	601	610	585	703	647	478	612
79	614	696	720	845	766	682	689	658	638	701
80	703	734	673	615	496	661	709	592	524	634
81	636	561	568	712	534	456	523	486	478	550
82	945	908	899	744	771	792	649	776	683	796
83	832	723	509	703	513	537	576	552	572	613
84	966	772	708	791	822	886	692	576	752	774
85	755	750	988	764	463	728	616	629	563	695
86	697	560	505	561	756	573	748	320	562	587
87	667	741	681	686	648	543	644	646	485	638
88	832	745	820	825	875	917	599	770	656	782
89	625	742	777	784	731	684	723	688	508	696
90	828	956	732	765	664	751	876	785	711	785
91	721	596	667	583	738	786	550	519	515	631
92	718	717	639	672	559	640	654	389	681	630
93	818	950	845	917	821	685	699	904	501	793
94	784	676	788	652	679	867	770	454	707	709
95	786	909	751	783	744	934	578	852	481	758
96	783	768	655	728	595	531	806	531	388	643
97	984	888	998	913	946	887	897	885	518	880
98	723	795	739	550	664	837	752	813	361	693
99	787	823	824	777	892	807	621	590	351	719
100	783	877	534	642	604	795	586	537	382	638

The numbers of item types in BR1v-BR15v are 3, 5, 8, 10, 12, 15, 20, 30, 40, 50, 60, 70, 80, 90, and 100, respectively. However, the average number of item types in the 100 new instances is 350. The number of orders in each of the 100 new instances is greater than the one in each case of BR1v-BR15v. As a result, **HCLPMC** takes longer times to solve the 100 instances than to solve BR1v-BR15v.

5. Conclusions

In this paper, we focus on the container loading problem of orders. Each order consists of at least one item. On any given day, certain orders are expiring, while other orders are not. The expiring orders must be loaded before the non-expiring ones. The items of an order must be entirely placed in the container or must be entirely left behind. The filling rate of the container should be maximized. To solve this problem, we propose a heuristic algorithm called **HCLPMC**, which consists of four steps. The expiring orders are handled in the first step and the second step, while the non-expiring ones are handled in the third step and the fourth step. In the first step (and the third step), a simulated annealing strategy is used to select a collection of orders, and in the second step (and the fourth step) a tree search procedure is used to pack the collection of orders into the container. We create 1500 cases (called BR1v-BR15v) from the popular BR1-BR15 instances to test **HCLPMC**. The mean filling rate of BR1v-BR15v by **HCLPMC** is 86.16%, whereas the mean computation time is 232 s. We also create 100 new instances to test **HCLPMC**. The mean filling rate of the new 100 instances by **HCLPMC** is 82.04%, whereas the mean computation time is 694 s. We will attempt to improve the filling rates of the cases in BR1v-BR15v and the new 100 instances in the future. Although several practical constraints are considered in our algorithms, we hope we can extend our work to include more constraints (such as weight and number of stacking layers) to provide more practical algorithms for container-loading problems.

Acknowledgement

The authors would like to thank the editors and the anonymous reviewers. This work was supported by the National Nature Science Foundation of China (Grant nos. 61104054, 61533019 and 71232006).

Appendix A

See Tables 10–13.

Appendix B. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cie.2017.04.021.

References

Bischoff, E. E. (2006). Three-dimensional packing of items with limited load bearings strength. *European Journal of Operational Research, 168*(3), 952–966.

Bischoff, E. E., Janetz, F., & Ratcliff, M. S. W. (1995). Loading pallets with non-identical items. *European Journal of Operational Research, 84*(3), 681–692.

Bischoff, E. E., & Marriott, M. D. (1990). A comparative evaluation of heuristics for container loading. *European Journal of Operational Research, 44*(2), 267–276.

Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega, 23*(4), 377–390.

Bortfeldt, A., & Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research, 131*(1), 143–161.

Bortfeldt, A., & Gehring, H. (1999). Zur Behandlung von Restriktionen bei der Stauraumoptimierung am Beispiel eines genetischen Algorithmus für das Containerbeladeproblem. In H. Kopfer & C. Bierwirth (Eds.), *Logistik management – Intelligente I + K technologien* (pp. 83–100). Berlin: Springer.

Bortfeldt, A., Gehring, H., & Mack, D. (2003). A parallel tabu search algorithm for solving the container loading problem. *Parallel Computing, 29*(5), 641–662.

Bortfeldt, A., & Mack, D. (2007). A heuristic for the three dimensional strip packing problem. *European Journal of Operational Research, 183*(3), 1267–1279.

Bortfeldt, A., & Wäscher, G. (2013). Constraints in container loading–A state-of-the-art review. *European Journal of Operational Research, 229*(1), 1–20.

Chen, C. S., Lee, S. M., & Shen, Q. S. (1995). An analytical model for the container loading problem. *European Journal of Operational Research, 80*(1), 68–76.

Davies, A. P., & Bischoff, E. E. (1999). Weight distribution considerations in container loading. *European Journal of Operational Research, 114*(3), 509–527.

Den Boef, E., Korst, J., Martello, S., Pisinger, D., & Vigo, D. (2005). Erratum to "the three-dimensional bin packing problem": Robot-packable and orthogonal variants of packing problems. *Operations Research, 53*(4), 735–736.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research, 44*(2), 145–159.

Egeblad, J., Garavelli, C., Lisi, S., & Pisinger, D. (2010). Heuristics for container loading of furniture. *European Journal of Operational Research, 200*(3), 881–892.

Egeblad, J., & Pisinger, D. (2009). Heuristic approaches for the two-and three-dimensional knapsack packing problem. *Computers & Operations Research, 36*(4), 1026–1049.

Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research, 141*(2), 393–409.

Eley, M. (2003). A bottleneck assignment approach to the multiple container loading problem. *OR Spectrum, 25*(1), 45–60.

Fanslau, T., & Bortfeldt, A. (2010). A tree search algorithm for solving the container loading problem. *INFORMS Journal on Computing, 22*(2), 222–235.

Fekete, S. P., Schepers, J., & Van der Veen, J. C. (2007). An exact algorithm for higher-dimensional orthogonal packing. *Operations Research, 55*(3), 569–587.

Gehring, H., & Bortfeldt, A. (1997). A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research, 4*(5–6), 401–418.

Gehring, H., & Bortfeldt, A. (2002). A parallel genetic algorithm for solving the container loading problem. *International Transactions in Operational Research, 9*(4), 497–511.

George, J. A., & Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research, 7*(3), 147–156.

He, K., & Huang, W. (2010). A caving degree based flake arrangement approach for the container loading problem. *Computers & Industrial Engineering, 59*(2), 344–351.

He, K., & Huang, W. (2011). An efficient placement heuristic for three-dimensional rectangular packing. *Computers & Operations Research, 38*(1), 227–233.

Hifi, M. (2002). Approximate algorithms for the container loading problem. *International Transactions in Operational Research, 9*(6), 747–774.

Hifi, M. (2004). Exact algorithms for unconstrained three-dimensional cutting problems: A comparative study. *Computers & Operations Research, 31*(5), 657–674.

Huang, W., & He, K. (2009). A caving degree approach for the single container loading problem. *European Journal of Operational Research, 196*(1), 93–101.

Junqueira, L., Morabito, R., & Sato Yamashita, D. (2012). Mip-based approaches for the container loading problem with multi-drop constraints. *Annals of Operations Research, 199*(1), 51–75.

Juraitis, M., Stonys, T., Starinskas, A., Jankauskas, D., & Rubliauskas, D. (2006). A randomized heuristic for the container loading problem: Further investigations. *Information Technology and Control, 35*(1), 7–12.

Lim, A., Ma, H., Qiu, C., & Zhu, W. (2013). The single container loading problem with axle weight constraints. *International Journal of Production Economics, 144*(1), 358–369.

Lim, A., Ma, H., Xu, J., & Zhang, X. (2012). An iterated construction approach with dynamic prioritization for solving the container loading problems. *Expert Systems with Applications, 39*(4), 4292–4305.

Lim, A., Rodrigues, B., & Wang, Y. (2003). A multi-faced buildup algorithm for three-dimensional packing problems. *Omega, 31*(6), 471–481.

Lim, A., Rodrigues, B., & Yang, Y. (2005). 3-D container packing heuristics. *Applied Intelligence, 22*(2), 125–134.

Liu, S., Tan, W., Xu, Z., & Liu, X. (2014). A tree search algorithm for the container loading problem. *Computers & Industrial Engineering, 75*, 20–30.

Liu, S., Zhao, H., Dong, X., & Cheng, C. (2016). A heuristic algorithm for container loading of pallets with infill boxes. *European Journal of Operational Research, 252*(3), 728–736.

Mack, D., Bortfeldt, A., & Gehring, H. (2004). A parallel hybrid local search algorithm for the container loading problem. *International Transactions in Operational Research, 11*(5), 511–533.

Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research, 48*(2), 256–267.

Morabito, R., & Arenalest, M. (1994). An AND/OR – Graph approach to the container loading problem. *International Transactions in Operational Research, 1*(1), 59–73.

Moura, A., & Oliveira, J. F. (2005). A GRASP approach to the container-loading problem. *IEEE Intelligent Systems, 20*(4), 50–57.

Parreno, F., Alvarez-Valdes, R., Oliveira, J. F., & Tamarit, J. M. (2007). A maximal-space algorithm for the container loading problem. *INFORMS Journal on Computing, 20*(3), 412–422.

Parreno, F., Alvarez-Valdes, R., Oliveira, J. F., & Tamarit, J. M. (2010). Neighborhood structures for the container loading problem: AVNS implementation. *Journal of Heuristics, 16*(1), 1–22.

Parreño, F., Alvarez-Valdés, R., Tamarit, J. M., & Oliveira, J. F. (2008). A maximal-space algorithm for the container loading problem. *INFORMS Journal on Computing, 20*(3), 412–422.

Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research, 141*(2), 382–392.

Ren, J., Tian, Y., & Sawaragi, T. (2011). A tree search method for the container loading problem with shipment priority. *European Journal of Operational Research, 214*(3), 526–535.

Wang, N., Lim, A., & Zhu, W. (2013). A multi-round partial beam search approach for the single container loading problem with shipment priority. *International Journal of Production Economics, 145*(2), 531–540.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research, 183*(3), 1109–1130.

Zhang, D. F., Peng, Y., & Leung, S. C. (2012). A heuristic block-loading algorithm based on multi-layer search for the container loading problem. *Computers & Operations Research, 39*(10), 2267–2276.

Zhang, D. F., Peng, Y., Zhu, W. X., & Chen, H. W. (2009). A hybrid simulated annealing algorithm for the three-dimensional packing problem. *Chinese Journal of Computers, 32*(11), 2147–2156.
Zhu, W., Huang, W., & Lim, A. (2012). A prototype column generation strategy for the multiple container loading problem. *European Journal of Operational Research, 223*(1), 27–39.
Zhu, W., & Lim, A. (2012). A new iterative-doubling Greedy-Lookahead algorithm for the single container loading problem. *European Journal of Operational Research, 222*(3), 408–417.

Liu Sheng. He received his Ph.D. degree in mechatronic engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2007. He is currently an associate research fellow of State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 100190, Beijing, China. His research area includes Combinatorial Optimization, Cutting and Packing. His emails are liusheng7801@163.com and sheng.liu@ia.ac.cn.

Shang Xiuqin. She is currently an assistant research of State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 100190, Beijing, China. Her research area includes intelligent Enterprise and Combinatorial Optimization. His email is xiuqin.zhao@ia.ac.cn.

Cheng Changjian. He is currently a senior engineer of Institute of Process Engineering, Chinese Academy of Sciences, 100190, Beijing, China. His research area includes intelligent enterprise and enterprise simulation. His email is changjian.cheng@ia.ac.cn.

Zhao Hongxia. She is currently an assistant research of State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 100190, Beijing, China. Her research area includes intelligent transportation and traffic simulation. His email is hongxia.zhao@ia.ac.cn.

Shen Dayong. He is currently an engineer of Qingdao Academy of Intelligent Industries, 266000, Qingdao, Shandong, China. His research area includes intelligent enterprise and enterprise simulation. His email is dayong.shen@qaii.ac.cn.

Wang Feiyue. He is currently the director of State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 100190, Beijing, China. His research area includes intelligent transportation and traffic simulation. His email is feiyuewang@gmail.com.