

# A Hybrid Simulated Annealing Algorithm for Container Loading Problem

Yu Peng

Department of Computer Science  
Xiamen University  
China

ypeng@cs.hku.hk

Defu Zhang

Department of Computer Science  
Xiamen University  
China

dfzhang@xmu.edu.cn

Francis Y.L. Chin

Department of Computer Science  
The University of Hong Kong  
Hong Kong

chin@cs.hku.hk

## ABSTRACT

This paper presents a hybrid simulated annealing algorithm for container loading problem with boxes of different sizes and single container for loading. A basic heuristic algorithm is introduced to generate feasible solution from a special structure called packing sequence. The hybrid algorithm uses basic heuristic to encode feasible packing solution as packing sequence, and searches in the encoding space to find an approximated optimal solution. The computational experiments on 700 weakly heterogeneous benchmark show that our algorithm outperforms all previous methods in average.

## Categories and Subject Descriptors

I.2.8 [ARTIFICIAL INTELLIGENCE]: Problem Solving, Control Methods, and Search – Heuristic methods

## General Terms

Algorithms

## Keywords

Container loading, Heuristics, Simulated annealing

## 1. INTRODUCTION

The problem addressed in this paper is a variant of packing problem. Packing problem has numerous applications in the cutting and packing industry. A good algorithm for packing problem is very important to save natural resource and minimize the trim loss.

Different optimal objective and loading constraints in practice lead to different variants of packing problems. An overview of the various types of packing and related cutting problem is described by Dyckhoff and Finke [5].

In this paper, three dimensional container loading problem (3D-CLP) is considered. It can be characterized as follows:

Given a rectangular container and a set of rectangular packing boxes, the objective is to determine a feasible arrangement of a subset of boxes which maximizes the volume of stowed boxes and meets the given loading constraints.

An arrangement is feasible if the following conditions are satisfied:

- Each stowed box is placed completely within the container.
- Each stowed box does not overlap another stowed box.
- Each stowed box is placed parallel to the container.

In practice, there are many constraints to meet in specific container loading problem. Here, only the following two constraints are included in the problem formulation.

### (C1) Orientation constraint

Orientation constraint means that only some specific sides of a box can be used as height. 3D-CLP without C1 can be converted to 3D-CLP with C1 by allowing every side to be used as height. Hence, we always consider 3D-CLP with C1.

### (C2) Stability constraint (Optional)

Stability constraint means that the stowed box must be fully supported by other stowed boxes or the bottom of the container. In other words, it forbids stowed boxes overhanging. Some applications in transportation industry require this constraint. In this paper, the algorithms for 3D-CLP with C2 and without C2 are both developed.

All boxes coincided in three dimensions and orientation constraints are considered to be the same type. A homogeneous box set is a set of boxes of one type. A weakly heterogeneous box set has few box types and many boxes per type. A strongly heterogeneous box set has a lot of box types and few boxes per type. Weakly heterogeneous box set is assumed in this paper.

In the recent years, many methods for the 3D-CLP have been developed. It is well known that the container loading problem is NP-hard (cf. [13]). Hence, the methods developed are heuristic approaches. Problem specific heuristics are proposed by Loh and Nee [9], Ngoi et al. [11], Bischoff et al. [2] and Bischoff and Ratcliff [1]. Intelligent graph search algorithms are introduced by Morabito and Arenales [10] and Pisinger [12], while Gehring and Bortfeldt [6] and Bortfeldt and Gehring [4] present genetic algorithms (GAs). Tabu search algorithms (TSAs) are given by Sixt [14] and Bortfeldt and Gehring [3]. Parallel tabu search algorithm is presented by Gehring and Bortfeldt [7], Bortfeldt and Gehring [16]. Parallel GA algorithm is proposed by Gehring and Bortfeldt [8]. Defu Zhang [17,18,19] also shows some interesting results by combining basic heuristic and simulated annealing algorithm.

## 2. BASIC HEURISTICS

### 2.1 The overall heuristics algorithm

The basic heuristic, which is an enhancement of the heuristics proposed by Bortfeld and Gehring [16], generates a feasible solution according to a given index vector named packing sequence. This heuristics algorithm is performed by loading a series so-called packing spaces in several phases. A packing space is an empty rectangular space ready to be loaded. While loading a packing space, only pre-defined structure called local arrangements, which will be mention later, of boxes are considered. The heuristics performs as follow: it keeps a list of available packing spaces and loads one packing space in each packing phase; at first, the container is the only packing space; in k-th phase, all possible local arrangements for the packing space are generated and sorted by certain criteria; then local arrangement specified by k-th element of packing sequence is chosen, boxes it contained are added to solution and residual packing spaces it generated are included as new available packing spaces which will be loaded latterly; when a phase ends, it pick another packing space in the available list to continue until all packing space are loaded.

### 2.2 Local arrangement

While generating local arrangement, only block structure is considered. A block is formed by boxes of the same type with same spatial orientation. Moreover, a block is a rectangular parallelepiped, and there is no space between boxes within it. A local arrangement contains one block, and the block is always placed in the left-rear-bottom corner of the packing space paralleled to the container.

Given a packing space, a box type and its orientation, a greedy filling algorithm is adopted to generating blocks. The greedy algorithm is based on the order of three dimensions; it fills the boxes as many as possible in the first dimension; then do the same thing in the second dimension; finally in the third. There are at most six possible filling methods for every box type. Figure 1 gives a boxes filling example with dimensions order  $\langle x, y, z \rangle$ .

It is noted that in the whole process, if there are no enough boxes, the filling process will stop immediately to make sure the used boxes do not exceed the limitation of problem specification.

After generation of the block, the unused part of the packing space must be divided into residual spaces to form a local arrangement. There exist many variant dividing methods; we only consider six simple dividing methods. First two dividing methods are adopted in 3D-CLP with C2, which are illustrated in Figure 2. These methods ensure that every generated residual packing space is supported by other stowed boxes or by the bottom of the container. The additional four dividing methods are adopted in 3D-CLP without C2, which are illustrated in Figure 3.

For a given packing space, all possible local arrangements will be generated. In this case, we need some criteria to sort them. We consider the local arrangement have bigger filled volume is better, and if tie the local arrangement have bigger residual volume is better.

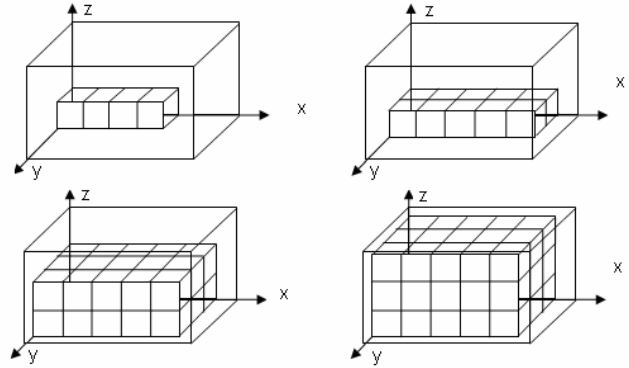


Figure 1. Example of boxes filling.

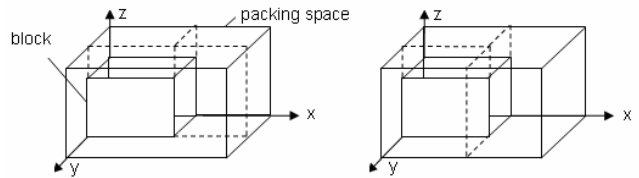


Figure 2. Stable dividing methods.

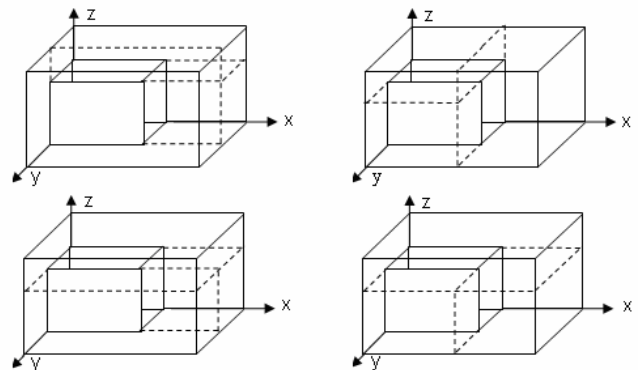


Figure 3. Additional dividing methods.

## 3. HYBRID ALGORITHM

As mentioned in last section, by using the basic heuristics, we can produce a feasible packing solution from a packing sequence. K-th element in packing sequence corresponds to local arrangement selection in k-th packing phase. Hence, a packing sequence is an encoding of a feasible packing solution, and searching in the space of packing sequences approximately corresponds to searching in the space of feasible packing solutions.

### 3.1 The neighborhood structure

The hybrid simulated annealing algorithm is carried out in the space of packing sequence. The neighborhood  $N(ps)$  of a packing sequence  $ps$  include all packing sequence  $ps'$ , for which the following applies:  $ps$  and  $ps'$  has same length but differs exactly one position  $j$ , namely  $ps[j] \neq ps'[j]$ , and for  $0 \leq i < \text{len}(ps)$ ,  $i \neq j$ , we have  $ps[i] = ps'[i]$ .

### 3.2 Complete hybrid algorithm

After constructing basic heuristics and neighborhood structure, we present the complete hybrid algorithm in Figure 4. In the

initialization, we select a packing sequence randomly; then we adopt simulated annealing algorithm to improve the utilization rate of solution found; controlling variables of this algorithm such as start and end temperature, annealing schedule, Markov chain length, etc. are got from input to increase the flexibility.

```

Initialize
Generate packing sequence ps randomly.
Calculate corresponding packing solution sln.
best := sln, t := start temperature

Simulated annealing algorithm
while t > end temperature do
  for i from 1 to L do
    Select ps' in N(ps) randomly.
    Calculate the solution sln' from ps' by heuristics.
    if sln'.v > sln.v then
      sln := sln', ps := ps'.
    else if random[0, 1) < exp((sln'.v - sln.v)/t) then
      sln := sln', ps := ps'.
    if sln.v > best.v then
      best := sln.
  Decrease t according to the annealing schedule
  
```

Figure 4. The complete hybrid algorithm.

### 3.3 Advance improvement

The setting of controlling variables may affect the performance dramatically. Computational experiments show that different settings of these variables perform better in certain cases. An improvement of this algorithm is that we can run this algorithm with different variable settings and select the best solution.

## 4. COMPUTATIONAL RESULTS

### 4.1 Applied test problems

The tests are based on well-known benchmark from Bischoff and Ratcliff [1]. The problems data can be found in OR-Library [15]. These problems are subdivided into seven test cases. The problems in the same test case have the same number of box types.

Orientation constraint (C1) is specified in the test problems. The hybrid algorithms with and without stability constraint (C2) are both tested.

### 4.2 Comparative tests

The algorithms were implemented in C++ and tests were run on an Intel Core Duo 2.0 GHz processor.

For the test problems from Bischoff and Ratcliff [1], results are available for the heuristics from Bischoff et al. [2], Bischoff and Ratcliff [1] and for the GAs from Gehring and Bortfeldt [6] and Bortfeldt and Gehring [4] and for sequential TSA and parallel TSA from Bortfeld and Gehring [16]. These results are summarized in Table 1. The results of the hybrid algorithms with C2 and without C2 are showed in Table 2 for comparison.

The results of comparative tests can be summarized as follows:

- ◆ The results reported by Bischoff and Ratcliff [1], Gehring and Bortfeldt [6] and Bortfeldt and Gehring [4] are based on full support of stowed boxes, namely with C2. The result reported by Bortfeld and Gehring [16] (both sequential and parallel) are based on 55% support of stowed boxes, which is a relaxed form of C2. For tests on full support of stowed boxes, Bortfeld and Gehring [16] only report the average volume utilization: 91.6% (sequential) and 92.2% (parallel).
- ◆ The hybrid algorithm for 3D-CLP with C2 achieves 92.83% volume utilization in average. This outperforms all previous works, even those algorithms which didn't consider C2.
- ◆ Along with the number of box types increasing, the running time of the algorithm increases monotonously. The reason is that the number of the possible local arrangements increases following the increase of box types, so does the running time.
- ◆ As the number of box types increases, the result utilization decreases almost monotonously. This is because the searching space expands quickly as the number of possible local arrangements increases in each packing phase. Therefore, it's harder to approximate optimal solution.
- ◆ The hybrid algorithm without C2 allows more possible local arrangements in each packing phase than that with C2. Hence, the former run longer but produce better solutions. Comparative tests show the former produce 0.67% better volume utilization rate than the latter in average.

## 5. CONCLUSION

In this paper, a hybrid simulated annealing algorithm for three dimensional container loading problem is proposed. A basic heuristic algorithm is introduced to encode feasible packing solution, and then the simulated annealing algorithm is applied to search in the encoding space. The comparative tests show that our algorithm outperforms all previous reported works. When relaxing the stability constraint, our algorithm achieves even better results in the standard test problems. The results prove that the neighborhood searching technique combined with a well-designed basic heuristic algorithm is a good way to solve container loading problem.

Table 1. The results of previous algorithms

Case	Bischoff et al. [2]	Bischoff, Ratcliff [1]	Gehring, Bortfeldt [6]	Bortfeldt, Gehring [4]	Bortfeld, Gehring [16]	Bortfeld, Gehring [16] (parallel)
BR1	81.76	83.79	85.80	87.81	93.23	93.52
BR2	81.70	84.44	87.26	89.40	93.27	93.77
BR3	82.98	83.94	88.10	90.48	92.86	93.58
BR4	82.60	83.71	88.04	90.63	92.40	93.05
BR5	82.76	83.80	87.86	90.73	91.61	92.34
BR6	81.50	82.44	87.85	92.72	90.86	91.72
BR7	80.51	82.01	87.68	90.65	89.65	90.55
Avg	82.0	83.5	87.5	90.1	92.0	92.7

**Table 2. The results of hybrid algorithms**

Case	Box types	With C2		Without C2	
		Time (s)	Rate (%)	Time (s)	Rate (%)
BR1	3	15.35	92.86	37.08	93.54
BR2	5	41.98	93.58	62.57	94.15
BR3	8	78.13	93.55	109.47	94.06
BR4	10	106.40	93.17	141.53	93.87
BR5	12	132.24	92.77	178.41	93.50
BR6	15	175.72	92.44	228.38	93.04
BR7	20	245.69	91.46	346.41	92.32
Avg	10	113.64	92.83	157.70	93.50

## 6. ACKNOWLEDGMENTS

This work was supported by the National Nature Science Foundation of China (Grant no. 60773126) and the Province Nature Science Foundation of Fujian (Grant no. A0710023)

## 7. REFERENCES

- [1] E.E. Bischoff, B.S.W. Ratcliff, Issues in the development of approaches to container loading, *Omega* 23 (1995) 377–390.
- [2] E.E. Bischoff, F. Janetz, M.S.W. Ratcliff, Loading pallets with non-identical items, *European Journal of Operational Research* 84 (1995) 681–692.
- [3] A. Bortfeldt, H. Gehring, Ein Tabu Search-Verfahren für Containerbeladeprobleme mit schwach heterogenem Kistenvorrat, *OR Spektrum* 20 (1998) 237–250.
- [4] A. Bortfeldt, H. Gehring, A hybrid genetic algorithm for the container loading problem, *European Journal of Operational Research* 131 (2001) 143–161.
- [5] H. Dyckhoff, U. Finke, *Cutting and Packing in Production and Distribution*, Physica, Heidelberg, 1992.
- [6] H. Gehring, A. Bortfeldt, A genetic algorithm for solving the container loading problem, *International Transactions in Operational Research* 4 (1997) 401–418.
- [7] H. Gehring, A. Bortfeldt, Ein verteilt-paralleles Tabu Search-Verfahren für Containerbeladeprobleme mit schwach heterogenem Kistenvorrat, in: P. Kall, H.-J. Lüthi (Eds.), *Operations Research Proceedings 1998*, Springer, Berlin, 1999, pp. 220–227.
- [8] H. Gehring, A. Bortfeldt, A parallel genetic algorithm for solving the container loading problem, *International Transactions in Operational Research* 9 (2002) 497–511.
- [9] T.H. Loh, A.Y.C. Nee, A packing algorithm for hexahedral boxes, in: *Proceedings of the Conference of Industrial Automation*, Singapore, 1992, pp. 115–126.
- [10] R. Morabito, M. Arenales, An AND/OR-graph approach to the container loading problem, *International Transactions in Operational Research* 1 (1994) 59–73.
- [11] B.K.A. Ngoi, M.L. Tay, E.S. Chua, Applying spatial representation techniques to the container packing problem, *International Journal of Production Research* 32 (1994) 111–123.
- [12] D. Pisinger, Heuristics for the container loading problem, *European Journal of Operational Research* 141 (2002) 143–153.
- [13] G. Scheithauer, Algorithms for the container loading problem, in: *Operations Research Proceedings, 1991*, Springer, Berlin, 1992, pp. 445–452. A. Bortfeldt et al. / *Parallel Computing* 29 (2003) 641–662 661
- [14] M. Sixt, *Dreidimensionale Packprobleme. Lösungsverfahren basierend auf den Meta-Heuristiken Simulated Annealing und Tabu-Suche*, Europäischer Verlag der Wissenschaften, Frankfurt am Main, 1996.
- [15] OR-Library, <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>
- [16] A. Bortfeldt, H. Gehring, D. Mack, A parallel tabu search algorithm for solving the container loading problem, *Parallel Computing* 29 (2003) 641–662.
- [17] ZHANG De-Fu, LI Xin. A Personified Annealing Algorithm for Circles Packing Problem, *ACTA AUTOMATICA SINICA*. 2005, 31(4):590-595.
- [18] ZHANG De-Fu, WEI Li-Jun, CHEN Qing-Shan, CHEN Huo-Wang. A Combinational Heuristic Algorithm for the Three-Dimensional Packing Problem. *Journal of Software*.2007,18(9):2083-2089.
- [19] Lijun Wei, Defu Zhang, and Qingshan Chen. A least wasted first heuristic algorithm for the rectangular packing problem. *Computers & Operations Research*, 2009,36(5):1608-1614.