

A New Prioritizing-Stacking Heuristic Algorithm for the Inner-City Truck Loading Problem

Safwan Altarazi¹

¹ Industrial Engineering Department, School of Applied Technical Sciences, German Jordanian University, Amman, Jordan

Correspondence: Safwan Altarazi, Industrial Engineering Department, School of Applied Technical Sciences, German Jordanian University, 35247 Amman, 11180, Jordan. Tel: 962-429-4522. E-mail: Safwan.altarazi@gju.edu.jo

Received: March 21, 2013

Accepted: April 22, 2013

Online Published: June 17, 2013

doi:10.5539/ijbm.v8n13p137

URL: <http://dx.doi.org/10.5539/ijbm.v8n13p137>

Abstract

Efficiently solving the truck loading problem (TLP) is crucial for achieving urban logistics objectives. Many heuristics, mainly based on wall-building and layer-building concepts, are presented in literature to solve this NP-hard problem. The current research presents a two-step novel prioritization-stacking heuristic algorithm for solving the TLP. The new algorithm primarily aims to improve the truck space utilization and reduce inventory costs at origin and destination points. In particular, the current work considers developing countries conditions in which demand is uncertain, cost minimization is prioritized over customer service, and routing optimization is complicated by a lot of traffic jams. The performance of the proposed heuristic is evaluated by applying it to a real-life case study from the Jordanian market. Additionally, the stacking part of the proposed heuristic is benchmarked with a literature data sample. Both implementations have proved the validity of the suggested heuristic algorithm.

Keywords: truck loading problem/container loading problem, urban logistics, prioritization of transported items

1. Introduction

Optimizing transportation activities is one of the most essential areas of logistics planning; it can greatly improve the overall logistics customer service level and reduce the total logistics cost. Truck loading/unloading is one of the key activities, especially for urban logistics where transportation of goods from warehouses to retailers and supermarkets is very frequent (Pan et al, 2011). A sound truck loading plan can increase the truck space utilization (TSU), improve transportation customer service level, increase energy efficiency, reduce inventory costs, and reduce transportation costs. However, truck load planning is a problem which involves numerous limiting factors, such as the complexity of the three-dimensional (3D) space, weight limitations and distribution within the truck space, shape of staked items and whether they are homogeneous or heterogeneous, loading and unloading sequence of the stacked items, etc.

In literature, the truck loading problem (TLP) fits under the more general container loading problem (CLP), (Ngoi et al, 1994; Bischoff & Ratcliff, 1995; Lim et al, 2012). Basically, the CLP requires completely stacking a group of 3D boxes into a 3D container without overlapping. This 3D loading problem is an NP-hard problem and is very difficult to solve in practice. Its theory is still in the initial stages of development and most of its optimization research is based on heuristics and metaheuristics (Lim et al, 2012). This paper contributes to the TLP research by presenting a novel two-step heuristic algorithm which aims to improve the TSU and reduce inventory costs at origin and destination points. Step one of the heuristic selects the items to be packed in the truck based on a multi criteria index, then, step two explains how the items should be packed within the 3D truck space. This heuristic specially/specifically applies to cases of: 1) high demand uncertainty, resulting in high inventory costs; 2) high priority of cost criterion, hence, reducing the cost (transportation and inventory costs) is more important than providing better service (expressed for example in low transportation cycle time), and 3) high complexity of routing optimization due to traffic jams; cases that are typically encountered in developing countries.

The rest of this paper is organized as follows. Section 2 briefly reviews the literature relating to the current research work. Section 3 provides a formal description of the problem. The suggested heuristic algorithm is

described in section 4. Section 5 includes the implementation and benchmarking of the suggested approach. Finally, the research conclusions and future extensions are summarized in section 6.

2. Literature Review

The container loading problem aims to efficiently stack 3D boxes in 3D containers. According to the optimized performance criteria and constraints, the CLP may have several variants including the knapsack loading, bin packing, and strip packing problems. The knapsack loading problem (Gehring et al, 1990; Pisinger, 2002) aims to load rectangular boxes into a rectangular container of fixed dimensions such that volume utilization is maximized. In the bin packing problem (BPP) (Martello et al, 2000; Faroe et al, 2003) the objective is to minimize the total number of containers (bins), or minimize the shipping cost, required to pack all items; containers can have fixed or varying dimensions. Finally, the strip packing problem (Bischoff & Marriott, 1990; Martello et al, 2003) assumes a container with fixed width and length but infinite height and aims to pack all items into the container such that the height is minimized.

Two main heuristics, the wall-building and layer-building, have been utilized to solve single container CLP, which resembles the TLP. The wall-building heuristic stacks items within regions created by walls across the container length or width (George & Robinson, 1980; Bischoff & Marriott, 1990; Martello et al, 2003); whereas, the layer-building approach loads items in a layer by layer sequence starting from the bottom of the container (Bischoff et al, 1995; Terno et al, 2000). Recently, many metaheuristic approaches have been applied to the CLP including: Genetic Algorithms (Bortfeldt & Gehring, 2001) and Tabu search (Bortfeldt & Gehring, 1998). Most the CLP/TLP literature adopts maximizing the volume utilization as the objective function. Practically, the CLP/TLP has various constraints; Wang et al (2009) summarized these constraints as follows: the non-overlapping constraint, the capacity constraint, the orthogonality constraint (items edges should be parallel to the truck edges), orientation constraint (items can only or cannot be rotated on the width-length plane), fragility constraint (non-fragile items can be stacked on top of each other but not on top of fragile items and fragile items can be stacked on top of non-fragile items only), stability constraint (minimum supporting area is required when items are stacked on top of each other), and sequential constraints (in the case of truck routing with multi-stops, unloading operation should not reshuffle items of upcoming customers in the route).

3. Problem Description

The considered inner-city truck loading problem can be described as follows: given 3D rectangular truck with a loading space (volume) defined by its Length L , width W , and height H , and a set of I 3D rectangular items to be stacked. If the items have length l_i , width w_i , height h_i , profit p_i , average demand d_i , and demand variability (standard deviation) s_i ; then the objective is to stack the truck in an appropriate way that will simultaneously optimize space utilization, profit, demand, and customer service. The followings constraints should be satisfied in solving the above problem:

- 1) All packed items must be totally included in the truck without overlapping.
- 2) The items are packed in parallel with the container sides.
- 3) The items have fixed height orientation but they can be rotated by 90° on the width-length plane.
- 4) Total items volume cannot exceed the truck space.

The positive corner of a 3D Cartesian coordinate system is used to represent the loading space of a truck with the L edge, W edge, and H edge respectively parallel to the x axis, y axis, and z axis. It is assumed that the truck has only one entrance represented by the z - y plane at L distance on the x -axis from the origin point. Figure 1 is an illustration of the 3D loading space of a truck.

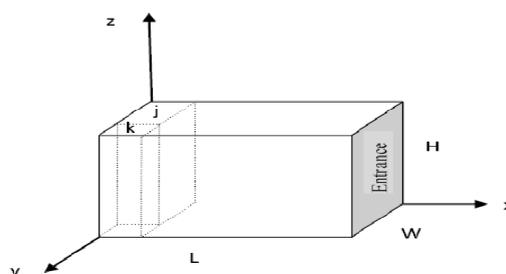


Figure 1. The truck's 3D loading space

4. The Two-Step Heuristic Algorithm

The suggested heuristic algorithm has two steps: prioritization and stacking. In the prioritization step the items are assigned a dimensionless weighted score based on the item's average demand, variability in demand, and profit. After prioritizing the items based on their scores, they are stacked according to the stacking rule in the second step of the heuristic algorithm. Before describing these two steps in details, it is worth mentioning that the presented heuristic algorithm has the following assumptions:

- 1) All items are non-fragile
- 2) All items have matching or identical densities
- 3) Loading and unloading activities are not considered
- 4) The largest item dimensions' are less than the truck dimensions; hence, in the worst case the largest item can singly be stacked in the truck.

4.1 The Prioritization Rule

The proposed prioritization rule contains three elements: item's average demand, demand variability, and profit. Each of the three elements is normalized, weighted, and then combined to produce a weighted average score for each item. The aim behind the rule is to prioritize the items to be loaded in the truck according to the aforementioned criteria. Below is a description of these elements.

1. Item's average demand: it is rational to start loading a truck with items which have the highest demand. Hence, it is required to rank the items in a descending order starting with the item having the highest demand. The normalized item's average demand score (N_{di}) is calculated as:

$$N_{di} = \frac{I - \text{rank}_i + 1}{I} \quad (1)$$

where:

I : is the number of items to be stacked

rank_i : is the demand-based descending rank of item i .

Note that as the items demand increases, the N_{di} increases with a score equals to 1 for the highest demanded item and a score of $1/I$ for the lowest demanded item.

2. Item's demand's variability: high uncertainty in an item's demand implies low customer service since the probability of finding the products in stock, that is, in the truck when demanded by the consumer, will be low. On the other hand, low stocking of uncertain products in the truck will reduce the percentage of unsold, hence returned, items. Therefore, the normalized item's demand variability score (N_{si}) is calculated as a ratio of the minimum coefficient of variance (CV) among all items (MinCV) to the coefficient of variance for item i (CV_i , $CV_i = s_i/d_i$). The largest N_{si} score, equals to 1, will be for the item with the lowest uncertainty (lowest CV_i). In contrast, the lowest value for the N_{si} will be close to zero. The N_{si} is given by the following equation:

$$N_{si} = \frac{\text{MinCV}}{CV_i} \quad (2)$$

3. Item's profit: the profit generated by an item is an essential element that should be included in the decision of what items to load a truck with. The normalized item's profit score (N_{pi}) is defined as a ratio of item i profit (p_i) to the maximum p_i for all items ($\text{Max}p_i$), as follows:

$$N_{pi} = \frac{p_i}{\text{Max}p_i} \quad (3)$$

After calculating the three above equations for all items, a weighted normalized score (Y_i) for each item is calculated as:

$$Y_i = w_d N_{di} + w_s N_{si} + w_p N_{pi} \quad (4)$$

where w_d , w_s , and w_p are weights of the above three elements. For determining these weights, expert's knowledge can be consulted.

It is worth emphasizing that Y_i will have values between 0 and 1 and that it is a normalized score since it is calculated from normalized elements. In addition, Y_i 's are actually calculated so that the sales (through N_{di}), customer service (through N_{si}), and profit (through N_{pi}) criteria are all simultaneously evaluated in a weighted

manner. After finding Y_i for all items the truck will be loaded starting with the items having the highest Y_i according to the stacking rule described in the next section.

4.2 The Stacking Rule

The suggested stacking rule can be categorized under the wall building heuristics family. By providing the rule with items' dimensions and their Y_i 's, and the truck dimensions (3D space); the rule starts by virtually dividing the 3D space into j ($j = 1, 2, \dots, J$) regions parallel to the y - z plane. Also, each of these j regions is divided into k ($k = 1, 2, \dots, K$) regions parallel to the x - z plane; j and k are initialized by very large values. The item with the highest Y_i is then selected to be stacked in the jk region (initialized as JK). If the item can be fitted in the jk region (through comparing the item's dimensions including its required number of boxes; with the jk region dimensions,) the item is stacked in the jk region starting from the coordination $(H - \sum_{b=1}^{b=B_i} bh_i, W - \sum_{b=1}^{b=B_i} bw_i, L - \sum_{b=1}^{b=B_i} bl_i)$, where B_i is the required number of boxes for item i . Then, the jk region empty space, the list of items required to be stacked, and the TSU are all updated. Otherwise (if the item cannot be fitted in the jk region,) the item is rotated around the x or y axes and the staking check is repeated. If rotation failed in stacking the item in the jk region, k is decreased by one and the staking check is repeated. The reduction of k regions continues until a capable jk region is found to stack item i , otherwise, j is decreased by one and the staking check is repeated. The reduction of j regions continues until a capable jk region is found to stack item i or the truck space is not capable anymore to stack any extra items. By the end of the first iteration of the algorithm a jk region should have been found to stack item i with the highest Y_i since it is assumed that the largest item should be contained within the truck's 3D space. Next (after successfully stacking the item with the highest Y_i in the first iteration of the algorithm,) the remaining items in I and the jk space are updated, and the truck utilization is calculated. The algorithm continues until all items are stacked or no extra items can be stacked because of truck's space limitation. The rule ends by providing what items have been stacked, their locations in the truck, and the TSU. Figure 2 summarizes the steps of the suggested stacking rule.

Before presenting the implementation of the proposed algorithm in the next section, it is worth mentioning that the C# (read as C sharp) programming language was used to code this algorithm.

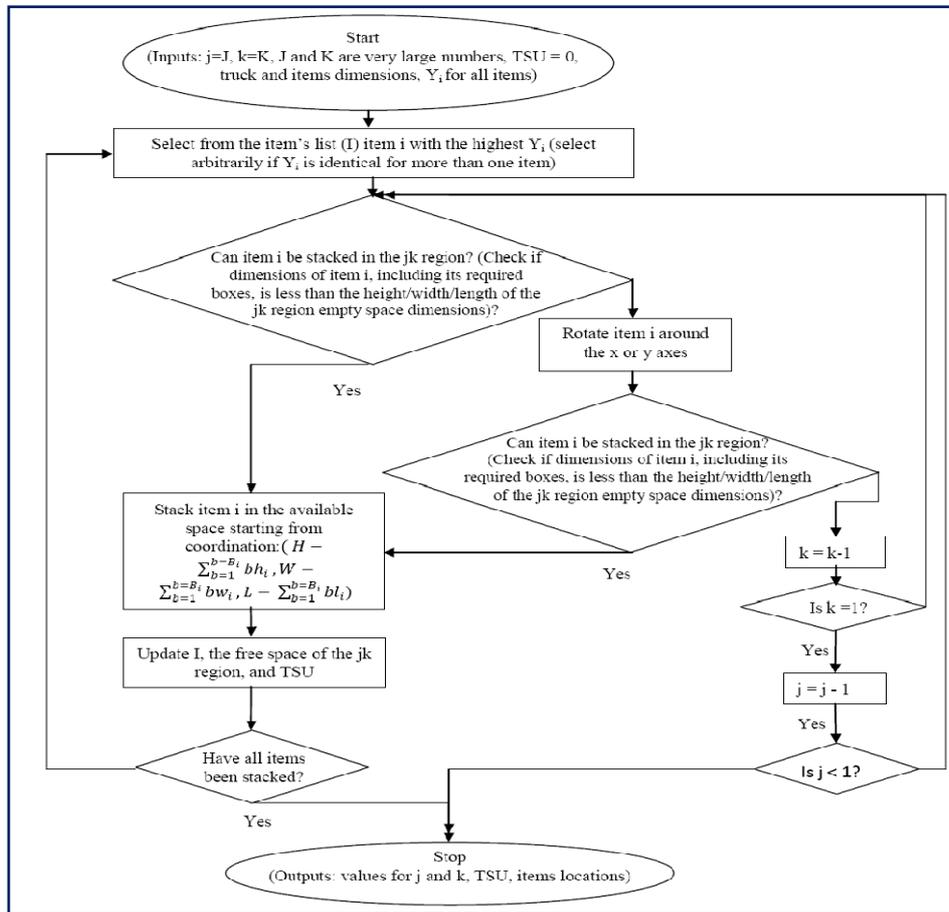


Figure 2. The flowchart of the suggested staking rule

5. Implementation

5.1 Case Study

The applicability of the proposed TLP prioritization-stacking heuristic algorithm is tested by applying it to an industrial company working in the Jordanian market. One of the company's routes was selected to perform the analysis. Although the total demand of the selected route consisted of 91 different items, only items of classes A (items representing 20% of cumulative percentage of all items after demand-based descending ranking of the items) and B (items representing the next 30% of the same demand-based descending ranking list) were included in the analysis. After conducting an ABC analysis it was found that class A had 18 different items representing about 74% of the total demand while class B had 26 items with 21% of the total demand. Table 1 presents the required information to apply the proposed heuristic for a sample of the 44 items (classes A & B). Considering the truck dimensions, they are: 2.00, 1.95, 5.95 meters for the height, width, and length, respectively.

Table 1. A sample of the items' information in the case study

Item	Rank	l (cm)	w (cm)	h (cm)	Average demand (Boxes, B _i)	Profit (p _i , \$)	CV
FTF	1	52	32	19	3	0.3	0.9
FTL	2	86	53	19	6	0.1	0.6
TTL	3	55	43	22	9	0.5	1.0
BDL	4	52	46	21	13	2.5	1.2
CDF	5	84	47	29	4	0.0	1.2
FDL	6	40	44	20	10	1.1	0.9
FTFS	25	73	38	22	12	0.1	2.0

BDB	26	66	56	14	4	0.4	2.4
NXT	27	28	24	18	2	0.1	3.7
NXTS	28	28	24	18	2	0.1	2.1
BDLI	29	52	46	21	13	2.1	2.2
TNF	41	36	22	16	2	0.4	2.9
NTN	42	28	24	18	1	0.0	2.0
BDBF	43	44	19	34	7	1.0	1.0
NTS	44	26	24	18	1	0.0	1.9

The input information was fed to the proposed heuristic algorithm. As explained in section 4, the heuristic key outputs are: ranking of the items, what items to stack in the truck, the location of each item in the truck demonstrated by the x-y-z coordination, and the TSU. After running the C# codes for this case study it was found that the described truck was able to stack all the 44 items with a TSU of 83.45%. Also, all items locations were decided by the algorithm output. It is worth mentioning that implementing the algorithm to other cases may result in stacking not all the required items but part of them; yet, items with highest priorities (highest Y_i 's) would be stacked.

5.2 Benchmarking

For validation purposes, the proposed heuristic algorithm, excluding the prioritization rule part, is compared with the CLP algorithm described in Pisinger (2002) using one of his samples data. Table 2 describes the problem as follows: 20 items/boxes are needed to be loaded in a container, the required number of boxes (the demand) is provided, and the boxes dimensions are given. Moreover, the container dimensions were 2.3 meters in height and width, and 5.9 meters in length-which is corresponding to the size of a standard 20 ft container.

By running the stacking part of the proposed heuristic algorithm using the data sample of table 2, the resulted space utilization was found to be 91.84% which is very close to the result found in (Pisinger, 2002). Indeed, this comparable performance of the stacking rule in the suggested heuristic algorithm validates the suggested methodology.

Table 2. The benchmarking data sample (Pisinger, 2002)

Box type	Width	Height	Depth	Number of boxes
1	85	60	26	5
2	36	69	28	7
3	68	26	46	6
4	71	100	100	5
5	28	31	42	6
6	109	29	51	8
7	94	52	43	9
8	32	106	113	6
9	52	115	45	13
10	29	68	44	9
11	42	41	115	12
12	97	59	48	8
13	106	53	40	5
14	105	68	32	3
15	107	43	110	3
16	39	62	80	6
17	29	56	73	7
18	31	47	42	10
19	39	41	114	8
20	101	38	40	10

6. Conclusions

This paper presented a novel heuristic algorithm for the TLP where the conditions of high demand uncertainty and cost criterion are dominating. After prioritizing the loading-requiring items based in their average demand,

demand variability and profit; step two of the algorithm stacks the items in the truck based on a proposed wall building heuristic algorithm. The validity of the prioritizing-stacking heuristic algorithm was accomplished through implementing it in a real-life case-study and benchmarking it with a literature problem. The suggested heuristic algorithm performed well in both instances. Future work will further address truck routing to enhance the applicability of the heuristic algorithm to more general real-life cases.

References

- Bischoff, E. E., & Marriott, M. D. (1990). A comparative evaluation of heuristics for container loading. *European Journal of Operational Research*, 44(2), 267-276. [http://dx.doi.org/10.1016/0377-2217\(90\)90362-F](http://dx.doi.org/10.1016/0377-2217(90)90362-F)
- Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega*, 23(4), 377-390. [http://dx.doi.org/10.1016/0305-0483\(95\)00015-G](http://dx.doi.org/10.1016/0305-0483(95)00015-G)
- 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. [http://dx.doi.org/10.1016/0377-2217\(95\)00031-K](http://dx.doi.org/10.1016/0377-2217(95)00031-K)
- Bortfeldt, A., & Gehring, H. (1998). Applying tabu search to container loading problems. In *Operations Research Proceedings 1997* (pp. 533-538). Springer Berlin Heidelberg.
- Bortfeldt, A., & Gehring, H. (2001). A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, 131(1), 143-161. [http://dx.doi.org/10.1016/S0377-2217\(00\)00055-2](http://dx.doi.org/10.1016/S0377-2217(00)00055-2)
- Faroe, O., Pisinger, D., & Zachariassen, M. (2003). Guided local search for the three-dimensional bin-packing problem. *INFORMS Journal on Computing*, 15(3), 267-283. <http://dx.doi.org/10.1287/ijoc.15.3.267.16080>
- Gehring, H., Menschner, K., & Meyer, M. (1990). A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research*, 44(2), 277-288. [http://dx.doi.org/10.1016/0377-2217\(90\)90363-G](http://dx.doi.org/10.1016/0377-2217(90)90363-G)
- George, J. A., & Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3), 147-156.
- 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. <http://dx.doi.org/10.1016/j.eswa.2011.09.103>
- Martello, S., Monaci, M., & Vigo, D. (2003). An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 15(3), 310-319. <http://dx.doi.org/10.1287/ijoc.15.3.310.16082>
- Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*, 48(2), 256-267. <http://dx.doi.org/10.1287/opre.48.2.256.12386>
- Ngoi, B. K. A., Tay, M. L., & Chua, E. S. (1994). Applying spatial representation techniques to the container packing problem. *The International Journal of Production Research*, 32(1), 111-123. <http://dx.doi.org/10.1080/00207549408956919>
- Pan, L., Chu, S. C., Han, G., & Huang, J. Z. (2011). A Heuristic Algorithm for the Inner-City Multi-Drop: Container Loading Problem. *International Journal of Operations Research and Information Systems (IJORIS)*, 2(3), 1-19. <http://dx.doi.org/10.4018/joris.2011070101>
- Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research*, 141(2), 382-392. [http://dx.doi.org/10.1016/S0377-2217\(02\)00132-7](http://dx.doi.org/10.1016/S0377-2217(02)00132-7)
- Terno, J., Scheithauer, G., Sommerweiß, U., & Riehme, J. (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, 123(2), 372-381. [http://dx.doi.org/10.1016/S0377-2217\(99\)00263-5](http://dx.doi.org/10.1016/S0377-2217(99)00263-5)
- Wang, F., Tao, Y., & Shi, N. (2009, April). A survey on vehicle routing problem with loading constraints. In *Computational Sciences and Optimization, CSO 2009*. International Joint Conference on (Vol. 2, pp. 602-606). IEEE.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).