Generating Stable Loading Patterns for Pallet Loading Problems.*

Waldemar Kocjan and Kenneth Holmström

Applied Mathematics, Mälardalen University, Västerås, Sweden

Abstract. This paper describes an integer programming model for generating stable loading patterns for the Pallet Loading Problem. The algorithm always gives optimal or near-optimal utilization of the pallet area and fulfills stability criteria for 98% of the test cases.

1 Introduction

A Pallet Loading Problem (PLP) is a problem of finding the optimal layout for packing a set of identical boxes onto a rectangular pallet. Normally the height of a box is considered to be fixed, and then a three dimensional pallet loading problem is reduced to a two dimensional problem $PLP(L, W, l, w)$. The objective is to allocate a maximum number of identical small rectangles of length l and width w on a bigger rectangle, a pallet, of length L and width W. The problem is a special case of the broader class of packing, cutting and placement problems as described in [7].

Typically, the PLP arises in logistics, where distributed goods have to be packed in layers on uniform pallets. The utilization of a pallet area is an important issue, which has an impact on the efficiency and cost of distribution. The stability of the generated loading pattern is a critical issue, yet not much studied. A solution of the PLP, which does not consider stability of a load, is more of theoretical value and hardly can be of use in practice.

In the current state-of-the-art methods for PLP, stability issues are either a secondary objective or not considered at all. Developers of methods that consider stability aspects often stress that there is a trade off between high utilization of a pallet and stability of the load pattern. The methods often work in a trial-and-error fashion generating a number of patterns to find the one which satisfy stability criteria best.

Our approach is different. We show that the stability criteria may be treated as the main objective without compromising utilization of a pallet. In contrast to previous methods using heuristic approaches to handle stability, our method is built on an Integer Programming (IP) formulation of the PLP, which has been shown to be very effective for moderate size problems (see eg. [1]). Furthermore, with the new method better results with respect to utilization and stability criteria are obtained on the test cases.

* Financed by Robotdalen (Robot Valley), http://www.robotdalen.org and Mälardalen University.

Section 2 gives a short description of the previous work on stability issues of the PLP. Section 3 describes the outline of our algorithm and the subsequent sections 4-7 give an in-depth description of each of its phases. Finally, Section 8 presents the results of the computational evaluation.

2 Related Work

The state-of-the-art on methods for solving PLP is given in e.g. [1]. Many of the methods use different types of heuristics. The block building heuristic is the most popular, see e.g. [3, 10, 11]. With this heuristic, boxes are allocated on the edges of the pallets and then the loading pattern is extended towards the center of the pallet.

Carpenter and Dowsland [5] introduce the following three stability criteria: The *supportive criterion* ensures interlock between columns of boxes in a loading pattern. The *base contact criterion* prevents situations where a box is not supported over an arbitrary percentage of its base. The *non–guillotine criterion* prevents creating patterns with straight cuts running across whole length or width of a pallet. The last criterion is closely related to the supportive criterion and is often considered redundant.

Bischoff [4] uses the criteria of [5] to verify stability of the loading pattern. To achieve maximum stability, Bischoff develops procedures of *compacting*, *centering blocks* and *disturbing gaps*, and incorporates them into the PLP solution algorithm by Bischoff and Dowsland [3].

Liu and Hsiao [9] consider a PLP, where boxes might be stacked on either their bottom, side or end surface, with the additional requirement of uniform height of each layer. The algorithm operates in five steps using the supportive criterion and the base contact criterion from [5]. A layout for each type of layer is computed using the block building heuristics of Smith and de Cani [11]. Then, new patterns are constructed by reflection and rotation. Finally, all combinations of generated patterns are constructed and the best stacking sequence is found. As in earlier stability methods, the loading pattern is an secondary objective compared to the utilization of a pallet.

3 The Algorithm for Generating Loading Patterns

The new algorithm for generating loading patterns works in three phases:

Phase 1 computes the maximal number of boxes and a maximal layout for one layer.

Phase 2 computes two layers such that both layer 1 placed on layer 2, and layer 2 placed on layer 1 fulfill (or nearly fulfill) the stability criteria.

Phase 3 creates full load by stacking boxes according to patterns computed in *Phase 2* up to an arbitrary height H.

Phase 2 computes optimal (or near-optimal) patterns and is the core of the algorithm with three stages. The algorithm proceeds from one stage to the next when it fails to produce a stable loading pattern. In contrary, if the generated pattern is stable, the algorithm jumps directly to the phase generating complete loading patterns. The three stages of *Phase 2* are the following:

Stage 1 computes new patterns by reflecting/rotating patterns computed in *Phase 1*.
Stage 2 computes a stable model using an IP formulation of the PLP, extended by stability constraints.
Stage 3 computes a near-optimal solution.

4 Computing the Optimal Layout of One Layer

This section and the following two describe the phases and stages of the new algorithm in detail. The optimal layout of a layer is computed using an IP formulation adopted from [2]. Two types of 0-1 variables are used, h_{ij} and v_{ij}. They equal 1 if a box is placed horizontally and vertically, respectively, with their lower left corner in position (i, j). The IP formulation of the PLP is

$$\max \sum_{i=0}^{L-l} \sum_{j=0}^{W-w} h_{ij} + \sum_{i=0}^{L-w} \sum_{j=0}^{W-l} v_{ij} \tag{1}$$

subject to

$$\sum_{i=\max\{0,r-l\}}^{\min\{r,L-l\}} \sum_{j=\max\{0,s-w\}}^{\min\{s,W-w\}} h_{ij} + \sum_{i=\max\{0,r-w\}}^{\min\{r,L-w\}} \sum_{j=\max\{0,s-l\}}^{\min\{s,W-l\}} v_{ij} \le 1 \tag{2}$$

$$(r = 0, \dots, L-1; s = 0, \dots, W-1),$$

$$h_{ij} \in \{0,1\} \quad (0 \le i \le L-l; 0 \le j \le W-w) \tag{3}$$

$$v_{ij} \in \{0,1\} \quad (0 \le i \le L-w; 0 \le j \le W-l). \tag{4}$$

Constraints (2) ensure that no boxes will overlap and are often referred to as *cover constraints*. The total number of variables is $L \times W \times 2$ and the number of constraints is $L \times W$. For a $PLP(100, 100, 11, 10)$ this leads to 20000 variables and 10000 constraints. These numbers can be significantly diminished by reducing the number of possible positions where boxes can be allocated. First, as shown in [8, 6], the number of points for a feasible placement of a box on a pallet can be reduced to *normal sets* given by the following equations,

$$S(L) = S(L, l, w) = \{r : r = \alpha l + \beta w, r + l \le L, \alpha, \beta \in \mathcal{Z}_+\} \tag{5}$$

$$S(W) = S(W, l, w) = \{r : r = \alpha l + \beta w, r + w \le W, \alpha, \beta \in \mathcal{Z}_+\}. \tag{6}$$

Reducing all points to normal sets for the $PLP(100, 100, 11, 10)$, the new problem size is 4140 variables and 2116 constraints, a significant reduction. Furthermore, using dominance relations adopted from dynamic programming by Scheithauer and Terno [10], define

$$\langle s \rangle_L := \max\{r \in S(L) : r \le s\}. \tag{7}$$

This leads to a further reduction of normal sets to *rasters* defined as

$$\widetilde{S}(L) = \widetilde{S}(L, l, w) = \{\langle L - r \rangle_L : r \in S(L)\} \tag{8}$$
$$\widetilde{S}(W) = \widetilde{S}(W, l, w) = \{\langle W - r \rangle_W : r \in S(W)\}. \tag{9}$$

For the $PLP(100, 100, 11, 10)$, applying this technique reduces the problem size to 612 variables and 324 constraints. The size of the problem after all reductions is about 3% of the original size of the problem! Additional reductions are possible by considering each orientation of a box separately.

5 Creating Layers by Simple Transformations

Given an optimal layer layout for a problem, create a subsequent layer using the following transformations:

1. a 180 degree rotation,
2. a reflection in the shorter pallet edge (W),
3. a reflection in the longer pallet edge (L).

The configurations created by combining the original and the generated patterns, in total six configurations, are evaluated with respect to base support and base contact criteria. Since the stability of layer 2 on layer 1, created by any of these transformations, implies stability of layer 1 placed on layer 2 ([5]), the complete load can be created by repeating stable patterns. Otherwise, the configuration closest to feasibility is used in the subsequent step as an initial solution.

6 Computing Stable Layers Using an IP Model

This section describes how a stable layout is computed by incorporating stability constraints into an IP formulation of the PLP in Section 4. The formulation includes variables denoting possible placement on two layers, the original layer on level $z = 0$ and a layer on the top of it with $z = t$, where t denotes the fixed height of a box. To assure that the number of boxes on each layer is optimal, add the constraint

$$\sum_{i=0}^{L-l} \sum_{j=0}^{W-w} h_{ij} + \sum_{i=0}^{L-w} \sum_{j=0}^{W-l} v_{ij} \geq opt \text{ for } z = 0, t \tag{10}$$

where *opt* is the optimal number of boxes at each layer as determined in the previous phase of the algorithm. Moreover, let M be an arbitrary large number. For each box at level $z = t$ in point (i, j), the supportive criterion is met by adding constraints

$$\forall (i,j) \sum_{m>i-l}^{m<i+l} \sum_{n>j-w}^{n<j+w} h_{mn} + \sum_{p>i-w}^{p<i+l} \sum_{q>j-l}^{q<j+w} v_{pq} \geq 2 + (1 - h_{ij}) \cdot M \tag{11}$$

$$\forall (i,j) \sum_{m>i-l}^{m<i+w} \sum_{n>j-w}^{n<j+l} h_{mn} + \sum_{p>i-w}^{p<i+w} \sum_{q>j-l}^{q<j+l} v_{pq} \geq 2 + (1 - v_{ij}) \cdot M \tag{12}$$

where (m, n) and (p, q) are points at $z = 0$. Constraints ensure that each box at level $z = t$ is supported by at least two boxes in the layer below.

Let ϵ be a value in $[0, 1]$, denoting the ratio of base area of a box in contact with boxes in the layer below. The base contact criterion for horizontally and vertically oriented boxes is assured by adding constraints (13) and (14), respectively, i.e.

$$\sum_{m>i-l}^{m<i+l} \sum_{n>j-w}^{n<j+w} ((\min(i+l, m+l) - \max(i, m)) \cdot (\min(j+w, n+w) - \max(j, n))) \cdot h_{mn}$$
$$+ \sum_{p>i-l}^{p<i+l} \sum_{q>j-w}^{q<j+w} ((\min(i+l, p+w) - \max(i, p)) \cdot (\min(j+w, q+l) - \max(j, q))) \cdot v_{pq}$$
$$\geq \epsilon \cdot l \cdot w \cdot h_{ij} \qquad (13)$$

$$\sum_{m>i-w}^{m<i+w} \sum_{n>j-l}^{n<j+l} ((\min(i+w, m+l) - \max(i, m)) \cdot (\min(j+l, n+w) - \max(j, n))) \cdot h_{mn}$$
$$+ \sum_{p>i-w}^{p<i+w} \sum_{q>j-l}^{q<j+l} ((\min(i+w, p+w) - \max(i, p)) \cdot (\min(j+l, q+l) - \max(j, q))) \cdot v_{pq}$$
$$\geq \epsilon \cdot l \cdot w \cdot v_{ij} \qquad (14)$$

To ensure that each box on the level $z = 0$ will maintain both stability criteria when placed on layer $z = t$, define the corresponding constraints for each box at layer $z = 0$. For boxes with $z = 0$ the stability criteria are fulfilled if at least two boxes are placed at its top, and also ϵ of its top area is covered by boxes at level $z = t$. In this formulation, problems which use rasters, as defined in Section 4, easily becomes overconstrained and require the addition of new possible placement points. To avoid such a situation, computation at this stage uses normal sets instead of rasters.

7 Near-Optimal Solutions

For most of the tested problems, a stable loading pattern is quickly generated at *Stage 1* or *Stage 2*. However, a few instances require a higher number of points for possible placement in order to produce a feasible solution. Some of the instances show a clear pattern: in the original, optimal layout of a layer there exists a waste space between packed boxes and edges of the pallet, and no waste in between boxes. For these cases, generate patterns which at each layer includes at most one box violating the base support criterion and with at least $l + w$ of the base area not supported by other boxes. The new layer is generated by duplicating the original layer and adjusting the position of each box with the least waste between packed boxes and the L and W edges of the pallet, and the integer value of the total box area divided by $l \cdot w$.

If no waste exists between boxes and edges of the pallet, do a new computation with the same stability constraints as the ones in Section 6, except that the lower bound on the number of boxes on each layer is diminished with an arbitrary amount.

8 Computational results

The method is evaluated on the set of benchmarks described in [9], but we investigate only configurations where the height of a box is fixed, corresponding to *B-type* of layers. The benchmarks for the problem are generated as follows. The pallet specifications are set to $L = 110$ cm, $W = 110$ cm and $H = 140$ cm. Boxes are generated with 1 cm increments, starting with $l = 30$ to 40 cm, $w = 20$ to 30 cm and height is set to 40 cm, which gives 121 box sizes. As in [9] we consider supportive criterion and base contact criterion with $\epsilon = 0.75$. Results of the evaluations are given in Table 1.

	Our evaluation	Reported in [9]
Fully stable patterns	95.8 %	Not reported
95% of boxes are stable	95.8 %	9.92 %
Average number of stable boxes	99.6 %	82.8 %
Optimal area utilization	98 %	Not reported

Table 1. Results of computational evaluation.

We conducted additional tests for Cover 1 problems, i.e. where each layer contains at most 50 boxes, $L \backslash W \leq 2$ and $l \backslash w \leq 4$. For 87.7%, optimal stable patterns were generated within a few minutes using a Dell Latitude D630 laptop.

References

1. Alvarez-Valdés, R., Parreño, F., Tamarit, J.M.: A branch-and-cut algorithm for the pallet loading problem. Comp. & OR **32** (2005) 3007–3029
2. Beasley, J.: An exact two dimensional non-guilatine cutting tree search procedure. OR **33** (1985) 49–64
3. Bischoff, E., Dowsland, W.: An application of the micro to product design and distribution. J. Opl Res. **33** (1982) 271–280
4. Bischoff, E.: Stability aspects of pallet loading. OR Spectrum **13** (1991) 189–197
5. Carpenter, H., Dowsland, W.: Practical consideration of the pallet loading problem. J. Opl Res. Soc. **36** (1985) 489–497
6. Christofides, N., Whitlock, C.: An algorithm for two-dimensional cutting problems OR **25** (1977) 30–44
7. Dyckhoff, H.: A topology of cutting and packing problems. Eur. J. Opl Res. **44** (1990) 145–159
8. Herz, J.: A recursive computational procedure for two-dimensional stock cutting. IBM J. Res. Dev. **16** (1972) 462–469
9. Liu, F.H.F., Hsiao, C.J.: A three dimensional pallet loading method for single-size boxes. J. Opl Res. Soc. **48** (1997) 726–735
10. Scheithauer, G., Terno, J.: The g4-heuristic for the pallet loading problem. J. Opl Res. Soc. **47** (1996) 511–522
11. Smith, A., de Cani, P.: An algorithm to optimize the layout of boxes in pallets. J. Opl Res. Soc. **31** (1980) 573–578