Heuristics for the Rectangle Packing Problem

Guntram Scheithauer Uta Sommerweiß *

August 1996

Abstract

In this paper the rectangle packing problem (RPP) is considered. The RPP consists in finding a packing pattern of small rectangles within a larger rectangle such that the area utilization is maximized,

We develop new heuristics for the RPP which are based on the G4-heuristic for the pallet loading problem. In addition to the general RPP we take also into account further restrictions which are of practical interest.

AMS Classification: 90C 10, 90C 05, 90B06

Key words: Packing Problems, Pallet Loading Problem, Cutting Problems, Heuristics

*Partially supported by BMBF, registration number 03-TE7DRE-8

1 Introduction

In this paper we consider the problem of determining a packing pattern of small not-necessary identical rectangles (so-called pieces) within a larger rectangle (so-called pallet) which maximizes the used area. This *rectangle packing problem* (RPP) is investigated in a lot of publications (cf. [5], [7], [8], [15], [16]) because of its large practical relevance in both the two-dimensional and three-dimensional case. Using the classification of DYCKHOFF [6], the considered problem is of type 2/B/O/F. The "2" indicates that a two-dimensional problem is considered. The objective is to pack a subset of the pieces ("B") within one larger region ("O") and we have relative few ("F") different types of pieces.

Since the well-known knapsack problem is involved in the packing problem, the latter also belongs to the class of $\mathcal{NP}$-hard problems. In fact, the number of different packing patterns increases in general exponentially with the number of pieces. Therefore, results for solving problems of medium size efficiently are reported in the literature mostly for heuristic algorithms.

The two-dimensional version of the well-known *pallet loading problem* (PLP) is a special case of the RPP where only identical pieces have to be packed. But the PLP by itself is of large practical interest and a lot of work is published in this field. We refer here exemplary to [4], [3], [2], [12] and [11] for various heuristics and exact algorithms. On the other hand there is only a little known with respect to the RPP.

Typically the PLP arises in a production process where uniform pallets (with identical products) have to be packed in a layer-wise manner (*manufacturer's PLP*, [10]). But in the distribution process the number of pieces of a certain type is restricted by order demands. As a consequence, non-uniform pallets have to be packed (so-called *distributor's PLP* [1]). Moreover, a layer-wise packing with identical pieces within a layer is not longer possible since the resulting (three-dimensional) packing patterns have in general a small utilization rate. Therefore, pieces of different type have to be combined in order to obtain a higher area utilization. If pieces of different height are combined, then the resulting structure of the (three-dimensional) packing pattern determines the combinations of further pieces which have to be packed above, etc. In this way, we are requested to make a compromise between the optimization of the area utilization, the structure complexity of the packing pattern and the available computation time.

In general, an optimal packing pattern of the RPP or a packing pattern obtained by any (general) algorithm has a very complicated structure which is in many cases not useful in practice. For that reason, our aim is to generate packing patterns having a special, more easy structure, namely a so-called k-block structure ([13]). The heuristics proposed in this paper are based on the approach used in [13] to compute optimal packing patterns with 4-block structure for the PLP. They consist in calculating of optimal packing patterns with respect to the k-block structure. In general, we consider only such patterns where identical pieces are arranged within any block. As a consequence we have in the three-dimensional case a constant height at least within a block.

In the following, we will give recurrence formulas to calculate optimal 4-block-patterns. In addition, we will regard additional restrictions such as limiting the number of different piece types in the pattern, lower and/or upper bounds pieces of a certain type have to or may be packed, and others. The usage of these heuristics is of benefit in the case where only a relative small number of piece types is given in difference to the well-known (two-dimensional) bin packing problem where a lot of different pieces have to be packed (cf. [7], [8]).

In order to formulate the RPP more precisely we use the following notations:

L	the length of the pallet (the large rectangle),
W	the width of the pallet,
m	the number of (different) types of pieces (the smaller rectangles),
I	$= \{1, \ldots, m\}$ the set of indices of piece types,
l_i	the length of a piece T_i of type i, $i \in I$,
w_i	the width of T_i, $i \in I$.

The objective of the RPP is to find a subset of the pieces which can be packed in a feasible manner within the pallet and which minimizes the unused area. A set of pieces is said to be packed feasibly if the pieces are orthogonally allocated, if they do not overlap each other and if they fit completely within the pallet. Some or all of the piece types may be rotated by 90 degrees. Without loss of generality we assume that all input data are positive integers.

The paper is organized as follows: In the next section we give some definitions and notations. In the main part (Section 3) we develop at first basic recurrence formulas and discuss several possibilities to save the computational amount. Then various additional restrictions are regarded in the following subsections. Remarks on the computation of more general patterns are given in Section 4. The efficiency of the new heuristics is shown by means of numerical experience with randomly generated instances in Section 5. Finally some concluding remarks are given.

2 Definitions and notations

A block or block pattern is an arrangement of one or several pieces within a rectangular region. In general we consider in the following only block patterns where identical pieces, i.e. pieces of one type, are packed. If all packed pieces in a block pattern have the same orientation we call it homogeneous.

In order to characterize the structure of a packing pattern of the whole pallet the following recursive definitions will be introduced.

Definition 1 *A packing pattern with n pieces of a pallet is said to be of* **guillotine structure (G-structure)** *if n is not greater than 3 or if there exists a partition of the pallet in two rectangles such that any of the two corresponding patterns is of G-structure by itself.*

Definition 2 *A homogeneous pattern is called to be of* **1-block structure**. *For $k \geq 2$, a packing pattern with n ($n \geq k$) pieces is said to be of* **k-block structure** *if there exists a partition of $I = \{1, \ldots, n\}$ in q disjunctive subsets I_j, $j = 1, \ldots, q$, $q \leq k$, such that each block pattern defined by I_j forms a pattern of p_j-block structure with $p_j \leq k$.*

According to this definition the set of packing patterns having k-block structure contains any packing pattern with a p-block structure for $p \leq k$.

Note, there is an important difference between the definition of patterns having a k-block structure and the *k-block patterns* used in the literature (cf. [12]) since for the latter it is required that each of the corresponding k block patterns consists of a homogeneous pattern.

Per definition, a packing pattern A of a pallet with n boxes is of n-block structure. But the question is: what is the smallest k for a given pattern A such that A is of k-block structure. Moreover, the following problem is unsolved so far even for the PLP: Given an instance (L, W, l, w) of the PLP. What is the smallest k such that the set of packing patterns having k-block structure contains an optimal pattern?

Assertion 1 *A packing pattern has a k-block structure with $k \leq 3$ if and only if the pattern is of G-structure.*

To see this one has only to make use of the recursive definition of G-structure and 3-block structure.

Because of the definition of the k-block structure, such a pattern with minimal waste can be computed using recurrence formulas of dynamic programming type. Therefore we need optimal patterns for smaller pallets (rectangles). As already mentioned we will consider in the following packing patterns of 4-block structure which contain in any of the at most 4 blocks only identical pieces.

Let $v_i(L', W')$ denote the area which is occupied by an optimal packing pattern with pieces of piece type i (i.e. with a maximal number of pieces) for a (rectangular) pallet $L' \times W'$.

It is well-known that the function v_i is monotonous non-decreasing and has staircase form. The step-off points are contained in $S_i(L) \times S_i(W)$ where

$$S_i(L) := \{r \in S_i \ : \ r \leq L\} \quad \text{and} \quad S_i := \{r \ : \ r = \alpha l_i + \beta w_i, \ \alpha, \beta \in Z_+\}.$$

In order to obtain efficient algorithms we use in fact instead of $v_i(L', W')$ the value which is obtained with the G4-heuristic for the PLP (cf. [13]) and which is optimal if $v_i(L, W) \leq 50$, and optimal in more than 99.9% if $v_i(L, W) \leq 100$.

For abbreviation, let

$$S(L) := \bigcup_{i \in I} S_i(L) \quad \text{and} \quad S := \bigcup_{i \in I} S_i.$$

These sets are defined due to the ideas of normalization (cf. Herz [9]) and of raster points (cf. [16]). Moreover, using

$$\langle s \rangle := \max\{r \in S : r \leq s\}$$

the reduced set $\widetilde{S}(L)$ of raster points is given by

$$\widetilde{S}(L) = \{\langle L - r\rangle : r \in S(L)\}$$

(cf. [16]). Obviously, it holds $\widetilde{S}(L) \subset S(L)$. For the reduced raster point set $\widetilde{S}(L)$ we define

$$\langle s\rangle_L := \max\{r \in \widetilde{S}(L) : r \leq s\}$$

It holds $\langle s\rangle_L \leq \langle s\rangle$ for any $s \geq 0$ and $\langle s\rangle_L = \langle s\rangle$ only if $\langle s\rangle \in \widetilde{S}(L)$.

The usage of $\widetilde{S}(L)$ realizes the application of the Nicholson principle of Dynamic Programming and is founded in that aspect that any $r \in S(L)$ for which an $r' \in S(L)$ exists with $r < r'$ and $\langle r\rangle_L = \langle r'\rangle_L$ can be omitted because of dominance considerations. Note, the definition of $\widetilde{S}$ leads in general to a smaller set in comparison to the approach of K. Dowsland (cf. [12], p. 45, e.g. PLP instance (65,65,13,11)).

In a similar way we define $S(W)$, $\widetilde{S}(W)$ and $\langle s\rangle_W$. Then, for $(L', W') \in S(L) \times S(W)$ (or $(L', W') \in \widetilde{S}(L) \times \widetilde{S}(W)$)

$$\overline{v}(L', W') := \max_{i \in I} \ v_i(L', W')$$

gives the maximal usable area of the pallet $L' \times W'$ if only pieces of one type are allowed to be packed. Let $i(L', W')$ denote the index of a piece type with $\overline{v}(L', W') = v_{i(L',W')}(L', W')$.

We define for $r > 0$

$$p_L(r) := \max\{s \in \widetilde{S}(L) : s < r\}, \qquad p_W(r) := \max\{s \in \widetilde{S}(W) : s < r\}$$

to be the predecessor of r with respect to $\widetilde{S}(L)$ or $\widetilde{S}(W)$, respectively. Note, only if $r > \langle r\rangle_L$ then we have $\langle r\rangle_L = p_L(r)$.

Moreover, let

$$I_L(s) := \{i \in I : S_i \cap [p_L(s) + 1, s] \neq \emptyset\}$$

and

$$I_W(s) := \{i \in I : S_i \cap [p_W(s) + 1, s] \neq \emptyset\}$$

be index-sets of suitable piece types with respect to length or width s, respectively.

3 Optimal patterns with 4-block structure

Within this section we develop recurrence formulas to compute an optimal packing pattern with respect to the set of patterns having 4-block structure and exactly one type of pieces per block. In the subsections several additional restrictions are considered.

Let $v(L, W)$ denote the value of a maximal pattern with 4-block structure and with respect to the actual considered additional restrictions.

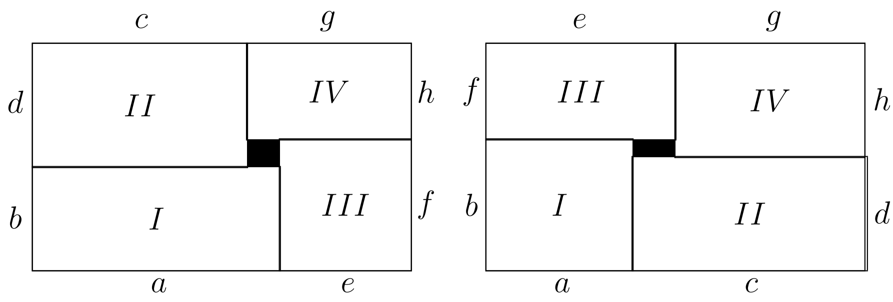


Figure 1: Notations for the recurrence formula

3.1 Basic recurrence

As it can be seen in Fig. 1, although we have two different cases, namely $a+g \geq L,\ b+h \leq W$ and $a+g \leq L,\ b+h \geq W$, which may lead to efficient packing patterns, it is sufficient to consider only one of the two cases because of symmetry arguments. Using the notations given in Fig. 1, left picture, we have in this case

$$v(L,W) := \max_{a\in\overline{S}_a} \max_{b\in\overline{S}_b} \left\{ \overline{v}(a,b) + \max_{c\in\overline{S}_c} \left\{ \overline{v}(c,d) + \max_{f\in\overline{S}_f} \{\overline{v}(e,f) + \overline{v}(g,h)\} \right\} \right\} \qquad (R1)$$

where

$$d := \langle W-b\rangle_W,\ e := \langle L-a\rangle_L,\ g := \langle L-c\rangle_L,\ h := \langle W-f\rangle_W, \qquad (1)$$

$$\overline{S}_a := \{s\in\widetilde{S}(L) : s \geq L/2\},\ \overline{S}_b := \widetilde{S}(W)\setminus\{0\}, \qquad (2)$$

$$\overline{S}_c := \{s\in\widetilde{S}(L) : s \leq a\},\ \overline{S}_f := \{s\in\widetilde{S}(W) : s \geq b\}. \qquad (3)$$

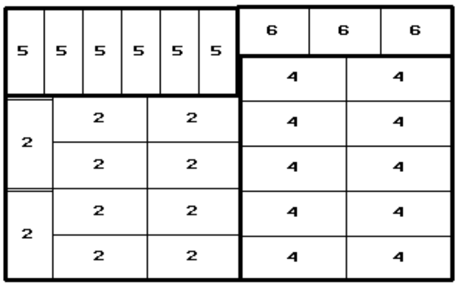
The restriction $a \geq L/2$ according to $\overline{S}_a$ results from $a+g \geq L$ and $a \geq g$. The definition of $\overline{S}_f$ is based on the aspect that any packing pattern of this 4-block structure with $f < b$ is dominated by packing patterns having a third block (III) with $f \geq b$.

In order to illustrate the influence of additional restrictions we will consider throughout this paper the following example defined in Table 1.

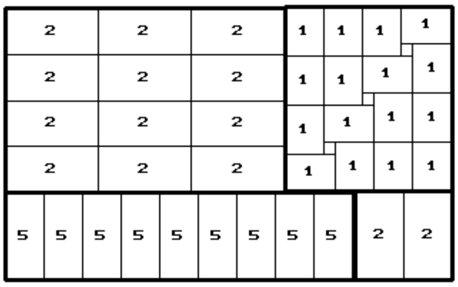
Table 1: Input data of the example

	pallet	piece 1	piece 2	piece 3	piece 4	piece 5	piece 6
length	1250	143	261	295	295	257	200
width	800	108	135	198	131	108	145

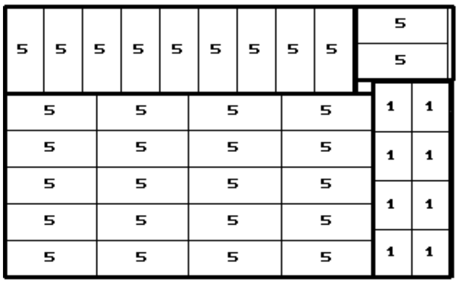
Figure 2 (a) shows an packing pattern which is optimal with respect to the set of patterns having 4-block structure.



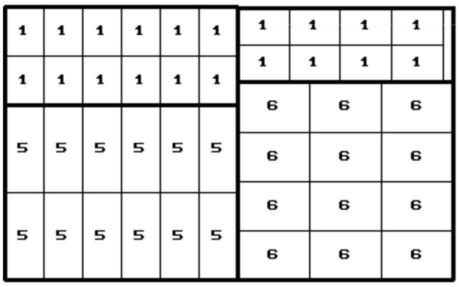
(a) v = 992336



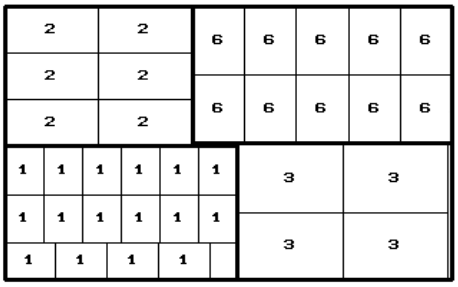
(b) v = 990198, t=3



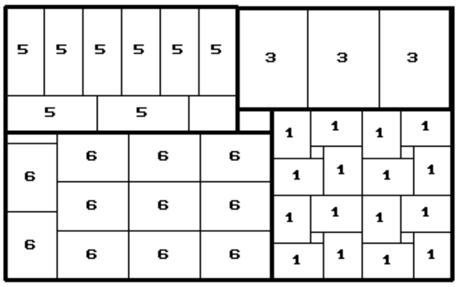
(c) v = 983988, t=2



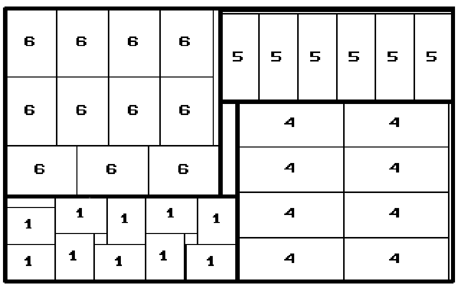
(d) v = 989952, t=3, connect=type 2



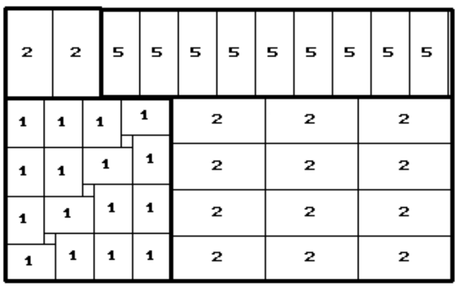
(e) v = 982154, $\bar{u} = (20, 9, 5, 8, 11, 11)$



(f) v = 963382, $\bar{u} = (20, 9, 5, 8, 11, 11)$, $\underline{u} = (0, 0, 0, 0, 0, 11)$



(g) v = 949136, $\bar{u} = (20, 9, 5, 8, 11, 11)$, $\underline{u} = (0, 0, 0, 8, 0, 11)$



(h) v = 990198, sequence=on

Figure 2: Packing patterns obtained for the same set of piece types but with various additional restrictions

In the cases $c = a$ and $f = b$, also patterns are considered which have a guillotine cut. If a vertical guillotine cut ($c = a$) is present then possibly a reduction of the computational amount can be reached. If the exchange of the two blocks I and II is allowed (which is true in the general case) then the point set $\overline{S}_c$ can be redefined to:

$$\overline{S}_c := \begin{cases} \{s \in \widetilde{S}(L) : s \le a\}, & b \le W/2, \\ \{s \in \widetilde{S}(L) : s < a\}, & W/2 < b < \langle W \rangle, \\ \{a\}, & otherwise. \end{cases} \tag{4}$$

Moreover, if the exchange of the blocks III and IV is feasible, then $\overline{S}_f$ can be reduced:

$$\overline{S}_f := \begin{cases} \{s \in \widetilde{S}(W) : s \ge b\}, & c < a, \\ \{s \in \widetilde{S}(W) : s \ge W/2\}, & otherwise. \end{cases} \tag{5}$$

Hence, if no further restrictions have to be regarded, recurrence formula (R1) can be used with parameters defined in (1), (2), (4) and (5).

Similar to [13] the amount for calculating $v(L, W)$ can be reduced with the help of upper bounds. In detail, realizing (R1) using four loops with increasing a-, b-, c- and f-values the following bounds can be applied (Let $\widetilde{v}(L, W)$ denote the optimal value for the corresponding subproblem defined by fixed parameters.):

1. If a is fixed then at least $s_1(L - a - e)$ units of area cannot be used where $s_1 := \min\{s \in S(W) : s > 0\}$ and $e := \langle L - a \rangle_L$. Hence,

$$\widetilde{v}(L, W) \le LW - s_1(L - a - e).$$

2. For a fixed a and b' with $b' \ge b$ it holds

$$\widetilde{v}(L, W) \le LW - b(L - a - e).$$

3. For fixed a and b it holds

$$\widetilde{v}(L, W) \le LW - (W - d)(L - e) + v_{i_1}(a, b),$$

where $d := \langle W - b \rangle_W$ and $i_1 = i(a, b)$.

4. For fixed a, b and c it holds

$$\widetilde{v}(L, W) \le LW - (W - d)(L - e) - (L - g)d + v_{i_1}(a, b) + v_{i_2}(c, d),$$

where $g := \langle L - c \rangle_L$ and $i_2 = i(c, d)$.

5. For fixed a, b, c and for f' with $f' \ge f$ it holds

$$\widetilde{v}(L, W) \le LW - (W - d)(L - e) - (L - g)d + v_{i_1}(a, b) + v_{i_2}(c, d) - (f + d - W)(g - e).$$

The advantage of these bounding criteria will be shown in Section 5 by means of numerical tests.

The recurrence formula (R1) makes use of the independence of piece types packed within different blocks. If such an independence does not occur because of some additional restrictions, then another recurrence can be used, namely

$$v(L,W) := \max_{a\in\overline{S}_a} \max_{b\in\overline{S}_b}$$

$$\max_{i_1\in I_1}\left\{v_{i_1}(a,b) + \max_{c\in\overline{S}_c}\max_{i_2\in I_2}\left\{v_{i_2}(c,d) + \max_{f\in\overline{S}_f}\max_{i_3\in I_3}\left\{v_{i_3}(e,f) + \max_{i_4\in I_4} v_{i_4}(g,h)\right\}\right\}\right\} \qquad (R2)$$

where

$$I_k := I, \quad k = 1,\ldots,4.$$

The block sizes d, e, g, h are defined in (1) and the point sets $\overline{S}_q$ with $q \in \{a, b, c, f\}$ are defined in (2), (3) or in (2), (4), (5).

Obviously, recurrence (R2) leads to the same result like (R1) but with an essential higher computational amount. (The increase of the running times is analysed in more detail in Section 5.) On the other hand, formula (R2) allows in a relative easy way the consideration of additional restrictions as will be seen below.

In order to save computational effort one can exploit dominance considerations. Let us consider a packing pattern with 4 blocks I, ..., IV as shown in Fig. 1, left. The length and width of the first block are denoted with a and b, respectively. Let us assume that $i_1 \notin I_W(b)$ holds. Then we have $v_{i_1}(a, p_W(b)) = v_{i_1}(a, b)$. Hence, the modified packing pattern with the reduced first block of width $p_W(b)$ has the same value. Therefore the index set I_1 can be reduced to

$$I_1^B := \{i \in I_W(b) : v_i(a, p_W(b)) < v_i(a, b)\}. \qquad (6)$$

A reduction of the point set $\overline{S}_c$ can be regarded in case $i_1 \notin I_L(a)$. In this case we have $v_{i_1}(p_L(a), b) = v_{i_1}(a, b)$. Therefore, all packing patterns with second block $c \times d$, $c \le p_L(a)$, (and first block $a \times b$, piece type i_1) cannot have a greater value than the corresponding packing pattern with the block $p_L(a) \times b$ and piece type i_1. Hence, in case $i_1 \notin \{i \in I_L(a) : v_i(p_L(a), b) < v_i(a, b)\}$ the point set $\overline{S}_c$ can be reduced to one element, i.e.

$$\overline{S}_c := \begin{cases} \{s \in \widetilde{S}(L) : s \le a\}, & i_1 \in \{i \in I_L(a) : v_i(p_L(a), b) < v_i(a, b)\},\ b < \langle W\rangle, \\ \{a\}, & \textit{otherwise}. \end{cases} \qquad (7)$$

In a similar way the sets I_2 and I_3 can be reduced:

$$I_2^B := \begin{cases} \{i \in I_L(c) : v_i(c, d) > v_i(p_L(c), d)\}, & b < \langle W\rangle,\ c < a, \\ I, & b < \langle W\rangle,\ c = a, \\ \emptyset, & \textit{otherwise}, \end{cases} \qquad (8)$$

$$I_3^B := \begin{cases} I, & f = b,\ c < a, \\ \{i \in I_W(f) : v_i(e, f) > v_i(e, p_W(f))\}, & \textit{otherwise}, \end{cases} \qquad (9)$$

If, in addition, the exchange of the blocks I and II is allowed then $\overline{S}_c$ can be redefined:

$$\overline{S}_c := \begin{cases} \{s \in \widetilde{S}(L) : s \leq a\}, & i_1 \in \{i \in I_L(a) : v_i(p_L(a), b) < v_i(a, b)\},\ b \leq W/2, \\ \{s \in \widetilde{S}(L) : s < a\}, & i_1 \in \{i \in I_L(a) : v_i(p_L(a), b) < v_i(a, b)\},\ W/2 < b < \langle W \rangle, \\ \{a\}, & \textit{otherwise}. \end{cases} \tag{10}$$

Furthermore, if the exchange of the blocks III and IV (in the case of a vertical guillotine cut) is feasible then $\overline{S}_f$ as defined in (5) can be used. Hence, if no further restrictions have to be taken into account, recurrence formula (R2) can be used with block sizes defined in (1), point sets defined in (2), (10) and (5), and index-sets defined in (6), (8), (9) and with $I_4^B := I$.

It is easy to see, if one of the index sets I_1^B, I_2^B (only if $c < a$) or I_3^B is empty then the corresponding subproblem (defined by fixed parameters a, b and c, respectively) can be fathomed because of dominance considerations.

In order to save computational amount the same fathoming criterion can be used as for recurrence (R1).

3.2 Restricted number of piece types

In this subsection we consider that case where the number of different piece types which are packed in a pattern with 4-block structure, is bounded by t with $t \in \{2, 3\}$. (In case $t = 1$ we have only to compare the v_i-values.) Note, because of the additional restrictions the placement of piece types within two blocks is no longer independent from each other. For that reason recurrence formula (R2) has to be used.

Let $t \in \{2, 3\}$. If a number of piece types has already been packed we have to take into consideration a reduction of the set of feasible piece types. This can be done with the help of the following index sets:

$$I_1^R := I,\ I_2^R := I,\ I_3^R := \begin{cases} I, & card(\{i_1, i_2\}) < t, \\ \{i_1, i_2\}, & \textit{otherwise}, \end{cases} ,$$

$$I_4 := \begin{cases} I, & card(\{i_1, i_2, i_3\}) < t, \\ \{i_1, i_2, i_3\}, & \textit{otherwise}. \end{cases}$$

Now, using

$$I_k := I_k^B \cap I_k^R, \quad k = 1, \ldots, 4,$$

recurrence (R2) leads to the desired result.

Considering the example given in the previous subsection, Fig. 2 (b) and (c) show optimal packing patterns with respect to the restriction $t = 3$ and $t = 2$, respectively.

3.3 Connectivity of packed piece types

One restriction for packing patterns which is of practical interest is as follows: If the same type of pieces is packed within more than one block, say for instance blocks p and q

($p, q \in \{I, II, III, IV\}$), then they have to be in "neighboured" blocks. Such restrictions are founded by the loading and re-loading process.

In a first case, such a "neighbourhood" is defined by $|p - q| = 1$. We refer it as variant 1 of connectivity. This restriction can be handled using again recurrence formula (R2) with help of suitable defined index sets:

$$I_1^C := I,\ I_2^C := I,\ I_3^C := (I \setminus \{i_1\}) \cup \{i_2\},\ I_4^C := (I \setminus \{i_1, i_2\}) \cup \{i_3\}.$$

Hence, the following index sets have to be used:

$$I_k := I_k^B \cap I_k^C, \quad k = 1, \ldots, 4.$$

Note, since the exchange of the blocks I with II or III with IV is not feasible, the point sets $\overline{S}_c$ defined in (7) and $\overline{S}_f$ defined in (3) have to be used.

Another kind of "connectivity" (referred to as variant 2) occurs when the blocks I and IV and the blocks II and III are defined to be not-connected. Then this form of restriction can be handled using again recurrence formula (R2) with appropriate index sets:

$$I_1^C := I,\ I_2^C := I,\ I_3^C := (I \setminus \{i_2\}) \cup \{i_1\},\ I_4^C := (I \setminus \{i_1\}) \cup \{i_2, i_3\}.$$

A third kind of "connectivity" (referred to as variant 3) occurs when only the blocks I and IV are defined to be not connected. Then the appropriate index sets are:

$$I_1^C := I,\ I_2^C := I,\ I_3^C := I,\ I_4^C := (I \setminus \{i_1\}) \cup \{i_2, i_3\}.$$

If additionally the number of different piece types which are arranged in the pattern, is bounded by t ($t \in \{2, 3\}$) then recurrence (R2) can also be used with the index sets:

$$I_k := I_k^B \cap I_k^R \cap I_k^C, \quad k = 1, \ldots, 4.$$

In Fig. 2 (d) there is shown an optimal packing pattern with respect to the additional constraints $t = 3$ and variant 2 of connectivity.

3.4 Upper bounds for pieces to be packed

Another restriction of practical interest which results e.g. from order demands is the following: Let $\overline{u}_i$, $i \in I$, denote an upper bound how often pieces of type i are allowed to be packed per pattern. In fact, only if $\overline{u}_i < v_i(L, W)$ we have a non-trivial restriction.

To handle this restriction we introduce a so-called pattern vector $p = (p_1, \ldots, p_m)^T \in Z_+^m$ where p_i gives the number pieces of type i have already been packed in the pattern. That means, if one index i of piece types is fixed within the recursion then p_i is increased by the corresponding number of packed pieces of this block.

Moreover, we define

$$\widetilde{v}_i(L, W) := \min\{v_i(L, W), \max\{(\overline{u}_i - p_i) l_i w_i, 0\}\}.$$

$\widetilde{v}_i(L,W)$ depends on p_i and hence, on the stage in the recursion.Furthermore, using

$$\overline{R} := \{i \in I : \overline{u}_i > p_i\}$$

we have $\widetilde{v}_i(L,W) = \min\{v_i(L,W), (\overline{u}_i - p_i)l_i w_i\}$ for $i \in \overline{R}$. Obviously, $\overline{R}$ depends also on the stage in the recursion.

Then such restrictions can be handled using recurrence formula (R3) which has the same structure like (R2):

$$v(L,W) := \max_{a\in\overline{S}_a} \max_{b\in\overline{S}_b}$$

$$\max_{i_1\in I_1}\left\{\widetilde{v}_{i_1}(a,b) + \max_{c\in\overline{S}_c}\max_{i_2\in I_2}\left\{\widetilde{v}_{i_2}(c,d) + \max_{f\in\overline{S}_f}\max_{i_3\in I_3}\left\{\widetilde{v}_{i_3}(e,f) + \max_{i_4\in I_4}\widetilde{v}_{i_4}(g,h)\right\}\right\}\right\} \qquad (R3)$$

where

$$I_k := I_k^B \cap \overline{R},\ \ k = 1,\dots,4,$$

and the block sizes d, e, g, h and sets $\overline{S}_q$ with $q \in \{a,b,c,f\}$ are defined in accordance to (R2).

The additional consideration of the other restrictions (upper bound for different piece types, "neighbourhood" condition) can be done simply by using the appropriate index-sets I_k, $k \in \{1,\dots,4\}$ and point sets $\overline{S}_q$, $q \in \{c,f\}$.

The packing pattern shown in Fig. 2 (e) is optimal with respect to the upper bounds $\overline{u} = (20,9,5,8,11,11)$. Since in the pattern Fig. 2 (a) the upper bound for the piece types 2 and 4 is violated a smaller area utilization results.

3.5 Lower bounds for pieces to be packed

In some cases a lower bound is given for the number pieces of a certain type have to be packed. With respect to the patterns considered here (which have to be of 4-block structure) we allow at most r such restrictions with $r \le 4$. Let $\underline{u}_i$, $i \in R \subset I$, $r = card(R) \le 4$, denote a lower bound pieces of type i have to be packed in the pattern. We assume for the sake of simplicity that such a packing pattern of 4-block structure exists.

In order to handle such restrictions using again recurrence formula (R2) we introduce

$$\underline{R} := \{i \in R : \underline{u}_i > p_i\}. \qquad (11)$$

Let again t denote the number of different piece types which are allowed to be packed. Hence, if this number is not restricted additionally, we have $t = 4$. Then we can define

$$I_k^L := \begin{cases} I, & card(\underline{R}) \le t-k, \\ \underline{R}, & otherwise, \end{cases} \quad k = 1,\dots,4. \qquad (12)$$

Using

$$I_k := I_k^B \cap I_k^L,\ \ k = 1,\dots,4,$$

the lower bounds can be handled.

In analogy to recurrence (R1) and (R2) the amount for calculating $v(L, W)$ can be reduced using corresponding upper bounds. In detail, realizing (R2) using four loops with increasing a-, b-, c- and f-values the following additional bounds can be applied. Let again $\widetilde{v}(L, W)$ denote the optimal value for the corresponding subproblem defined by fixed parameters. Then:

- For fixed a, b and i_1 it holds if

  $$LW - (L-e)(W-d) < \sum_{i \in \underline{R}} (\underline{u}_i - p_i) l_i w_i$$

  then the pattern is not feasible. Continue with the next i_1.

  If $card(\underline{R}) = 4$ then the pattern does not become feasible. Continue with the next i_1.

- For fixed a, b, c and i_2 it holds if

  $$LW - (L-e)(W-d) - d(L-g) < \sum_{i \in \underline{R}} (\underline{u}_i - p_i) l_i w_i$$

  then the pattern is not feasible. Continue with the next i_2.

  If $card(\underline{R}) \geq 3$ then the pattern does not become feasible. Continue with the next i_2.

- For fixed a, b, c, f and i_3 it holds if

  $$gh < \sum_{i \in \underline{R}} (\underline{u}_i - p_i) l_i w_i$$

  then the pattern is not feasible. Continue with the next i_3.

  If $card(\underline{R}) \geq 2$ then the pattern does not become feasible. Continue with the next i_3.

- At last, the feasibility of the complete packing pattern generated has to be verified.

The additional consideration of the other restrictions (upper bound for different piece types, "neighbourhood" condition) can be done by using appropriate index sets I_k, $k \in \{1, \ldots, 4\}$ and point sets $\overline{S}_q$, $q \in \{c, f\}$.

3.6 Desired numbers for packed pieces

The restriction that for some piece types $i \in R$ $(1 \leq card(R) \leq 4)$ a given number u_i of pieces have to be packed can be handled as follows: For any $i \in R$ we define the lower bound $\underline{u}_i := u_i$ and the upper bound $\overline{u}_i := u_i$ and combine the considerations of the previous two subsections. Hence, using recurrence formula (R3) with the index sets I_k defined in (12) yields the desired result.

Fig. 2 (f) gives an optimal packing pattern with 11 pieces of type 6 where some other upper bounds are regarded. If in addition 8 pieces of type 4 also have to be packed then the packing pattern in Fig. 2 (g) is the best with 4-block structure.

This situation is a special case of that when some lower bounds $\underline{u}_i$, $i \in R$, ($card(R) \leq 4$) and some (nontrivial) upper bounds $\overline{u}_i$ are given.

If additionally the number of piece types is bounded by t ($t \in \{2,3\}$, $card(R) \leq t$) then the following index sets have to be used in (R3):

$$I_k := I_k^B \cap I_k^R \cap I_k^L \cap \overline{R}, \quad k = 1, \ldots, 4,$$

In order to save computational amount suitable tests can be used in analogy to previous subsections. The feasibility of the patterns generated has to be verified.

If in addition the connectivity has to be regarded then easily the following index sets have to be used for all of the variants of connectivity:

$$I_k := I_k^B \cap I_k^R \cap I_k^L \cap \overline{R} \cap I_k^C, \quad k = 1, \ldots, 4,$$

3.7 Sequential conditions

In some cases sequential conditions of the following form occur: If pieces of a certain type i are already packed then it is only allowed to pack pieces of type $k \in P_i$ where $P_i \subset I$. Such kind of restrictions may arise when a given packing sequence has to be regarded, e.g. if the pieces have different height and they have to be packed with decreasing height.

Taking into account such restrictions, it is not possible to employ symmetry arguments. For that reason the set $\overline{S}_a$ has to be redefined and for both cases, shown in Fig. 1, a recurrence formula has to be applied. In detail, for patterns with $c \leq a$ (notations according to Fig. 1, left) we have

$$\overline{S}_a := \widetilde{S}(L) \setminus \{0\}. \tag{13}$$

Then these conditions can be handled using recurrence formula (R2) with the enlarged point sets $\overline{S}_a$ (defined in (13)), $\overline{S}_c$ (defined in (7)), $\overline{S}_f$ (defined in (3)), and the index-sets

$$I_k := I_k^B \cap \overline{P}_k, \ k = 1, \ldots, 4,$$

where

$$\overline{P}_1 := I, \ \overline{P}_k := \overline{P}_{k-1} \cap P_{i_{k-1}}, \ k = 2, 3, 4.$$

For the second case (right pattern in Fig. 1) a similar recurrence has to be used.

If it is demanded that the pieces have to be packed with increasing type index then the pattern in Fig. 2 (h) arises.

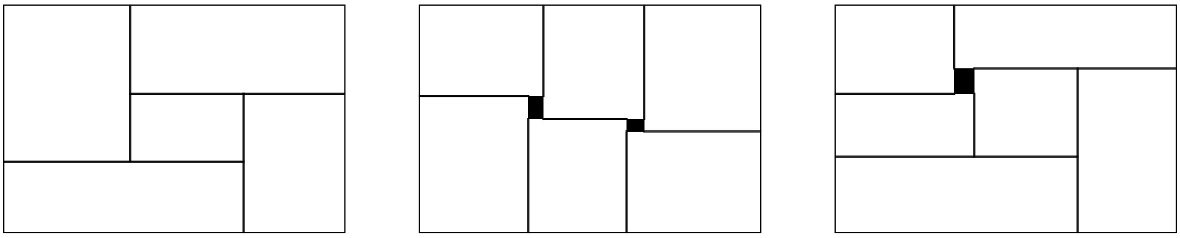


Figure 3: Patterns with 5- and 6-block structure

4 More general packing patterns

4.1 Remarks on patterns with k-block structure and $k > 4$

The proposed heuristic to generate packing patterns with 4-block structure can easily be extended to compute 5- or 6-block patterns. Such a structure can be reached if one of the blocks is replaced by two new blocks. Hence, one obtains 5-block structure from 4-block structure and 6-block structure can be obtained from 5-block structure. It is to remark that there also exist patterns (cf. Fig. 3) with a structure which cannot be generated in this way. To calculate such patterns appropriate recurrence formulas have to be used.

Patterns of a k-block structure with a k greater than 4 may be especially of interest when the upper bounds are relative small and more than 4 piece types are allowed to be packed within a pattern.

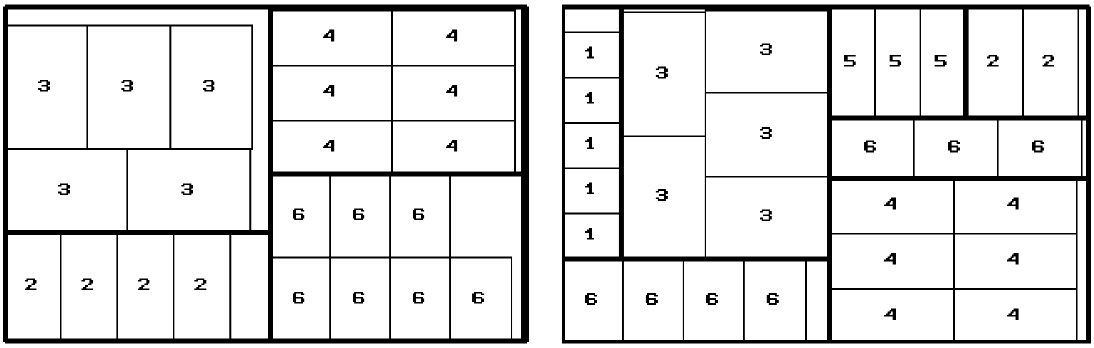
4.2 General packing patterns

If we go without the restriction that only pieces of one type occur in a block, we obtain more general packing patterns. Especially, if for any $L' \leq L$ and any $W' \leq W$ the values $v(L', W')$ are computed in similarity to the basic recurrence packing patterns can be obtained with an area utilization rate which is in general higher than that obtainable in the case considered above. On the other hand, a very "non-ordered" packing pattern will be obtained in general which is not of advantage for practical considerations. Furthermore, the computational amount increases rapidly. Moreover, the handling of additional restrictions as considered above is much more difficult and expensive.

For that reason another possibility to generate more general packing patterns looks more suitable. If some or all of the four blocks are considered to contain at most two types of pieces, packing patterns containing up to eight different piece types may be generated. In detail, instead of $\overline{v}(a, b)$ in the basic recursion one can use

$$\max \left\{ \max_{0 \leq l \leq a/2} \overline{v}(l, b) + \overline{v}(a - l, b), \ \max_{0 \leq w \leq b/2} \overline{v}(a, w) + \overline{v}(a, b - w) \right\},$$

that is, the pattern with pieces of one type is possibly replaced by a 2-block pattern with two (different) types of pieces.



(a) v = 867860, 4-block pattern

(b) v = 957878, 8-block pattern

Figure 4: 4- and 8-block pattern for the example with $\underline{u} = (0, 0, 0, 0, 0, 7)$, $\overline{u} = (5, 4, 5, 6, 3, 7)$

In order to illustrate such improvements we consider our example with $\underline{u} = (0, 0, 0, 0, 0, 7)$ and $\overline{u} = (5, 4, 5, 6, 3, 7)$. The two patterns in Fig. 4 are obtained with the 4-block heuristic and an 8-block heuristic which is a direct generalization of the 4-block heuristic allowing 2-block patterns in any of the 4 blocks. (Block I contains 4 pieces of type 6; II and III contain pieces of type 1, 3 and 4, 6, respectively; and the remaining pieces are in block IV.)

5 Computational experience

In order to investigate the computational behaviour of the proposed algorithms numerical tests with randomly generated instances were carried out.

For any instance we used $L = 1250$ and $W = 800$. Therefore the total area used in a packing pattern gives also the percentage of area utilization. In a first series of test instances the lengths and widths of the pieces are chosen randomly in the range [100,300] according to a uniform distribution. Therefore, any generated piece fits at least 8 and at most 96 times on the pallet.

Within our numerical tests we generated 100 sets of instances with $m = 4, \ldots, 10$ piece types. Thereby the instances with $m = 4$ piece types are chosen independently from each other. But the further (fifth, sixth, etc.) piece type are added successively to obtain the instances with $m > 4$.

Table 2 gives the results obtained with respect to the area utilization. Any entry is an average value of 100 instances.

The first row in Table 2 denoted by "G4-heuristic" contains the average area utilization (in %) of the G4-heuristic which is in principle the optimal value of the standard two-dimensional PLP.

The next row "4b-heuristic" gives the area utilization obtained with the new heuristic allowing up to 4 different piece types per packing pattern. Although we considered relative

Table 2: Average area utilization (in %) for range [100,300]

m	4	5	6	7	8	9	10
G4-heuristic	96.6	96.9	97.2	97.3	97.5	97.6	97.7
4b-heuristic	99.0	99.2	99.4	99.5	99.6	99.6	99.7
t=3	99.0	99.2	99.3	99.5	99.5	99.6	99.6
t=2	98.7	98.9	99.1	99.2	99.3	99.3	99.4
co=2, t=3	98.9	99.1	99.3	99.4	99.5	99.5	99.6
up. bounds	96.0	97.2	97.8	98.2	98.4	98.6	98.8
exact=1	95.0	96.1	96.7	97.0	97.3	97.5	97.7
exact=2	93.2	94.2	94.8	95.1	95.3	95.5	95.7
sequ=on	98.9	99.1	99.3	99.4	99.5	99.5	99.6
average	97.4	97.9	98.2	98.4	98.5	98.7	98.8

Table 3: Average running time (in sec.) for range [100,300]

m	4	5	6	7	8	9	10
rec 1	0.20	0.34	0.54	0.73	1.02	1.37	1.78
rec 2	0.74	1.30	2.04	2.79	3.83	5.20	6.72
t=3	0.76	1.31	2.05	2.85	3.90	5.21	6.78
t=2	0.87	1.43	2.29	3.17	4.30	5.48	6.95
co=2, t=3	0.82	1.40	2.21	3.09	4.21	5.59	7.24
up. bounds	1.88	3.10	4.83	6.87	9.87	13.53	18.14
exact=1	2.44	4.22	6.77	10.04	14.68	20.54	27.75
exact=2	2.25	3.78	5.79	8.27	11.57	15.39	20.35
sequ=on	1.83	2.83	4.11	5.53	7.00	9.01	11.37
average	1.45	2.42	3.76	5.33	7.42	9.99	13.16

small pieces (which can be packed at most 96 times) a remarkable increase of area utilization is reached.

The next two rows "$t = 3$" and "$t = 2$" indicate the behaviour if the number of different piece types allowed to be packed is restricted. The row "co=2, $t = 3$" gives the results when additionally to the restriction "$t = 3$" a connectivity constraint (here variant 2) has to be regarded. But the decrease of area utilization is not larger than 0.1% in average.

In order to investigate the influence of upper bounds we assigned to any piece type an upper bound which equals a third of its G4-value. Hence, the algorithm is forced to arrange at least 3 different piece types in a packing pattern. The row "up. bounds" shows the decreased values of area utilization.

In addition the lower bound of one (row "exact=1") or two (row "exact=2") piece types is set equal to the upper bound. The piece types which have to be packed in such a fixed number are always the same in the corresponding set of instances (increasing m). Hence, these pieces may be combined with an increasing number of further piece types.

Table 4: Average area utilization (in %) for range [150,300]

m	4	5	6	7	8	9	10
G4-heuristic	94.9	95.7	96.2	96.5	96.7	96.8	97.1
4b-heuristic2	98.4	98.8	99.0	99.2	99.3	99.4	99.4
t=3	98.4	98.7	99.0	99.1	99.3	99.3	99.4
t=2	98 0	98.3	98.6	98.7	98.9	99.0	99.1
co=2, t=3	98.3	98.6	98.9	99.1	99.2	99.3	99.4
up. bound	93.9	95.5	96.4	97.0	97.4	97.7	97.9
exact=1	92.8	94.2	94.9	95.5	95.8	96.0	96.3
exact=2	90.8	92.1	92.8	93.3	93.7	93.8	94.0
sequ=on	98.3	98.6	98.9	99.1	99.2	99.3	99.4
average	96.1	96.9	97.3	97.6	97.9	98.0	98.1

Table 5: Average running time (in sec.) for range [150,300]

m	4	5	6	7	8	9	10
rec 1	0.06	0.12	0.18	0.26	0.36	0.5	0.66
rec 2	0.26	0.46	0.72	1.04	1.47	2.02	2.70
t=3	0.27	0.47	0.73	1.05	1.46	2.01	2.64
t=2	0.28	0.48	0.73	1.02	1.36	1.79	2.31
co=2, t=3	0.29	0.51	0.79	1.14	1.59	2.14	2.78
up. bounds	0.56	1.00	1.59	2.44	3.61	5.03	6.73
exact=1	0.67	1.21	2.00	3.11	4.73	6.64	9.00
exact=2	0.60	1.00	1.56	2.23	3.18	4.32	5.59
sequ=on	0.60	0.94	1.35	1.81	2.45	3.21	4.13
average	0.44	0.76	1.18	1.73	2.48	3.40	4.49

In order to simulate a different height of the pieces and the resulting sequential conditions we simply required that the piece types packed have to have increasing indices for increasing block number. Row "sequ=on" shows the little decrease of area utilization which is not larger than 0.1% in average.

Altogether, as also row "average" indicates, a high area utilization can be obtained also if additional constraints have to be regarded.

The average CPU times (in seconds, PC 586, 200 MHz) are given in Table 3 in similarity to Table 2. (Thereby the time needed to calculate the G4-values $v_i(L', W')$ is not contained since this can be done a priori.) In addition we give a comparison of the times needed with recurrence 1 and recurrence 2, rows "rec. 1" and "rec. 2", respectively. As expected, there arises an increase of running time but recurrence 2 does not take more than 4 times in comparison to recurrence 1.

The increase of running time in the rows "up. bounds", ..., "sequ=on" is founded in that aspect that the bounding criteria used in the algorithms do not exploit these additional

Table 6: Comparison of bounding criteria (average running time in sec.)

bound. crit.	4	5	6	7	8	9	10
1–5	0.12	0.23	0.36	0.60	0.90	1.19	1.52
–	2.63	6.7	14.31	27.27	53.55	88.14	124.89
1	2.59	6.64	14.2	27.07	53.15	87.52	124.04
2	2.8	7.08	15.04	28.60	55.93	91.94	129.83
3	0.97	2.07	3.63	6.63	11.06	14.67	18.36
4	0.37	0.74	1.24	2.03	3.41	4.72	6.16
5	0.28	0.6	1.05	1.74	2.88	4.22	5.58
2–5	0.12	0.22	0.36	0.59	0.89	1.17	1.52
1, 3–5	0.12	0.24	0.37	0.59	0.91	1.20	1.49
1,2,4,5	0.26	0.57	0.96	1.60	2.66	3.87	5.13
1–3, 5	0.14	0.25	0.37	0.60	0.92	1.24	1.56
1–4	0.23	0.42	0.66	1.07	1.70	2.10	2.59
1–5	0.21	0.36	0.55	0.81	1.12	1.49	1.90
3–5	0.21	0.35	0.54	0.81	1.13	1.50	1.91

conditions. Here further investigations are required.

Nevertheless, the absolute average computation times indicate the applicability of the algorithms.

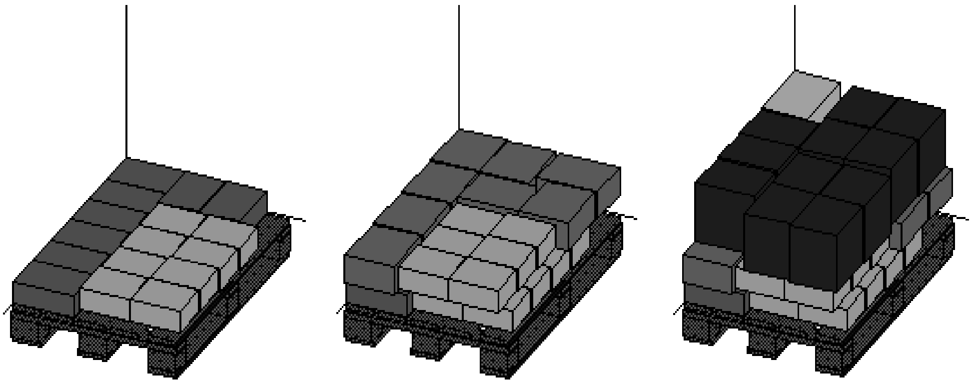
In order to deepen the insides in the numerical behaviour a second series of test instances was computed. In this series only the range of the piece sizes is reduced to [150,300] so that the pieces fit at most 40 times on the pallet. The results are shown in Tables 4 and 5.

Again a remarkable increase of the area utilization is obtained in comparison to the G4-values (Table 4, rows "4b-heuristic" and "G4-heuristic"). Since (in average) larger pieces have to be packed fewer computational amount is needed. Only approximately 30% of running time is used in comparison to the first series (Table 5).

In order to investigate the efficiency of the bounding criteria described in Subsection 3.1 we compared several combinations of these five tests for recurrence 1. The first column in Table 6 gives the information which of the tests is used. The entries of Table 6 are average values (in seconds) of 10 instances with the exception of the last two rows where average values of 100 instances are given. Since there is no absolute dominating test we prefer the variant with all bounding tests.

6 Concluding remarks

In this paper the problem of calculating optimal packing patterns of small rectangles on a pallet is considered. We propose new heuristics which are based on the 4-block structure of packing patterns. Since, in general, for practical applications additional restrictions have to be regarded, we developed a recurrence formula for which it is relative easy to



(a) Layer 1 (b) Layer 2 (c) Layer 3

Figure 5: Applying the 4-block heuristics for three-dimensional pallet loading

handle such conditions. This is shown by means of some concrete requirements of practical interest. Moreover, it should not be very difficult to take into account further restrictions in this way.

Depending on the concrete circumstances when applying these heuristics further modifications may be required. The pictures in Fig. 5 illustrate such a situation. During the computation of the packing pattern for the second (third) layer the height structure of the previous layer as well as remaining pieces of already used types have to be considered. A new heuristic approach is proposed in [14] for solving the (three-dimensional) multi-pallet loading problem which uses the heuristics described in this paper.

References

[1] E. E. Bischoff and M. S. W. Ratcliff. Issues in the development of approaches to container loading. *OMEGA*, 23(4):377–390, 1995.

[2] H. Carpenter and W. B. Dowsland. Practical considerations of the pallet–loading problem. *J. Oper. Res. Soc.*, 36(6):489–497, 1985.

[3] K. A. Dowsland. The three–dimensional pallet chart: An analysis of the factors affecting the set of feasible layouts for a class of two–dimensional packing problems. *J. Oper. Res. Soc.*, 35(10):895–905, 1984.

[4] K. A. Dowsland and W. B. Dowsland. A comparative analysis of heuristics for the two–dimensional packing problem. In *Paper for EURO VI Conference*, 1983.

[5] K. A. Dowsland and W. B. Dowsland. Packing problems. *European J. Oper. Res.*, 56:2–14, 1992.

[6] H. Dyckhoff. A typology of cutting and packing problems. *European J. Oper. Res.*, 44:145–160, 1990.

[7] H. Dyckhoff and U. Finke. *Cutting and packing in production and distribution.* Physica Verlag, Heidelberg, 1992.

[8] H. Dyckhoff, G. Scheithauer, and J. Terno. Cutting and packing: An annotated bibliography. Preprint MATH-NM-08-1996, Techn. Univ. Dresden, 1996. to appear in *Combinatorial Optimization: An Annotated Bibliography,* eds. Dell'Amico, Maffioli, Martello.

[9] J. C. Herz. Recursive computational procedure for two-dimensional stock cutting. *IBM J. of Research and Development*, 16:462–469, 1972.

[10] T. J. Hodgson. A combined approach to the pallet loading problem. *IIE Transactions*, 14(3):175–182, 1982.

[11] G. Naujoks. *Optimale Stauraumnutzung.* (Diss.) Deutscher Universitätsverlag, Wiesbaden, 1995.

[12] J. Nelißen. *Neue Ansätze zur Lösung des Palettenbeladungsproblems.* (Diss. RWTH Aachen) Verlag Shaker, Aachen, 1994.

[13] G. Scheithauer and J. Terno. The G4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, 47:511–522, 1996.

[14] G. Scheithauer, J. Terno, J. Riehme, and U. Sommerweiß. A new heuristic approach for solving the multi-pallet packing problem. Technical report, MATH-NM-03-1996, TU Dresden, 1996.

[15] P. E. Sweeney and E. L. Ridenour. Cutting and packing problems: A categorized, application–oriented research bibliography. Technical report, Working Paper #610, School of Business Administration, University of Michigan, 1989.

[16] J. Terno, R. Lindemann, and G. Scheithauer. *Zuschnittprobleme und ihre praktische Lösung.* Verlag Harri Deutsch, Thun und Frankfurt/Main, 1987.

Authors

Guntram Scheithauer, Uta Sommerweiß

Institute of Numerical Mathematics

Technical University Dresden

D - 01062 Dresden, Mommsenstr. 13, Germany

e-mail: scheit@math.tu-dresden.de, sommerw@math.tu-dresden.de