Consecutive ones matrices for multi-dimensional orthogonal packing problems

Cédric Joncour, Arnaud Pêcher

Noname manuscript No.
(will be inserted by the editor)

Consecutive ones matrices for multi-dimensional orthogonal packing problems ⋆

Cédric Joncour · **Arnaud Pêcher**

Received: 1 July, 2010 / Accepted: ???

Abstract The multi-dimensional orthogonal packing problem (OPP) is a well studied decisional problem. Given a set of items with rectangular shapes, the problem is to decide whether there is a non-overlapping packing of these items in a rectangular bin. The rotation of items is not allowed. A powerful caracterization of packing configurations by means of interval graphs was recently introduced. In this paper, we propose a new algorithm using consecutive ones matrices as data structure. This new algorithm is then used to solve the two-dimensional orthogonal knapsack problem. Computational results are reported, which show its effectiveness.

Keywords orthogonal packing problem · interval graph · consecutive ones matrices

1 Introduction

The two-dimensional orthogonal packing problem (2d-OPP) is a well studied NP-hard optimization problem [16, 21, 27]. Given a set of items $\mathscr{I}$ with rectangular shapes (each item $i \in \mathscr{I}$ has a width w_i and an height h_i), the problem is to decide whether the set of items is *feasible*, that is whether there is a non-overlapping packing in a rectangular bin of width W and height H. Rotation of items is not allowed (see Figure 1). The two-dimensional orthogonal knapsack problem (2d-OKP) is to compute the maximum value of a feasible set: if every item $i \in \mathscr{I}$ has a positive value p_i, the aim is to exhibit a feasible subset of items $\mathscr{I}' \subset \mathscr{I}$ such that $\sum_{i \in \mathscr{I}'} p_i$ is maximal (see Figure 1).

Many authors gave MIP formulations introducing a discretization of the space [3, 11, 4, 7], or relying on the relative position of items [10, 23, 24, 25]. Caprara and Monaci [8] gave a comparative study of several Branch & Bound algorithms solving strip packing problem. Baldacci and Boschetti proposed a cutting plane approach to solve this problem [1]. Constraint programming was also used to provide exact algorithms [5, 6, 15, 22, 25, 26]. Dual feasible functions were also used by several authors (see [14] for a recent survey).

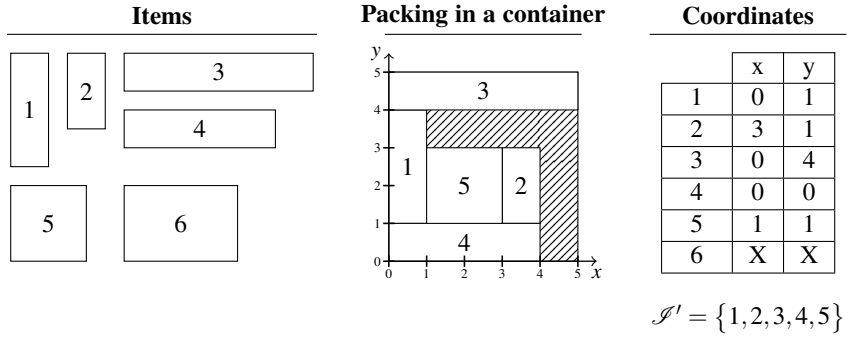
We consider the D-dimensional orthogonal packing problem and the D-dimensional orthogonal knapsack problem (with $D \geq 2$). Fekete and Schepers [18, 19] introduced a new approach using graph theory. Their algorithm is one of the fastest, even in the two dimensional case. They used some tuple of interval graphs as data structures to store a feasible packing. Though Fekete and Schepers' s algorithm is very efficient, there are sill symmetry issues. We propose a new algorithm using some consecutive ones matrices as data structures, due to Fulkerson and Gross's characterization of interval graphs. Our approach is able to eliminate some of these symmetry issues.

This paper is organized as follows. In the first section, we introduce the model used to check if a set of items has a feasible packing. We describe also the algorithm to check feasibility and we exhibit several optimizations to handle symmetry issues. In the second section, we describe how the knapsack problem is solved, and give computational results with respect to standard benchmarks.

⋆ This research was partially supported by the ANR Project GraTel ANR-blan-09-blan-0373-01, 2010-2012.

C. Joncour
IMB / INRIA Bordeaux - Sud-Ouest, University of Bordeaux,
351 Cours de la Libération, 33405 Talence Cedex, France
E-mail: cedric.joncour@math.u-bordeaux1.fr

A. Pêcher
IRIT, University of Toulouse,



	x	y
1	0	1
2	3	1
3	0	4
4	0	0
5	1	1
6	X	X

$\mathscr{I}' = \{1,2,3,4,5\}$

Fig. 1: A feasible selection of items for 2d-OKP with $p_i = 1, \forall i \in \mathscr{I}$

2 A new algorithm to solve orthogonal packing

2.1 Consecutive ones matrices for orthogonal packing

Let n denote the number of items and D be the dimension of the Euclidean space. Let $\mathscr{I} = \{1,\dots,n\}$ be a set of items. For every $d \in \{1,\dots,D\}$ and $i \in \mathscr{I}$, let w_i^d be the width of item number i w.r.t. dimension d. For every $d \in \{1,\dots,D\}$, let W^d be the width of the bin w.r.t. dimension d. The set of items $\mathscr{I}$ is feasible (see Figure 1) if there is a tuple of coordinates $(x_i^1,\dots,x_i^D) \in \mathbb{R}_D^+$ for every item $i \in \mathscr{I}$ such that:

$$\forall i \in \mathscr{I}, \forall d \in \{1,\dots,D\} : x_i^d + w_i^d \leq W^d, \tag{1}$$

$$\forall i,j \in \mathscr{I} (i \neq j),\ \exists d \in \{1,\dots,D\} : [x_i^d, x_i^d + w_i^d) \cap [x_j^d, x_j^d + w_j^d) = \emptyset. \tag{2}$$

We denote by feasible packing, a set of tuple of coordinates of a feasible set of items satisfying the constraints (1) and (2). A feasible packing verifies touching assumption if every item is immediately to the "right" of an other item or touches the "left border" (w.r.t. every dimension):

$$x_i^d \in \{0\} \cup \left\{ x_j^d + w_j^d : j \in \mathscr{I} \setminus \{i\} \right\}, \quad \forall i \in \mathscr{I}, d \in \{1,\dots,D\}. \tag{3}$$

Lemma 1 *For every feasible set of items $\mathscr{I}$, there is an associated feasible packing which verifies touching assumption.*

Proof Let $(x_i^1,\dots,x_i^D)_{i\in\mathscr{I}} \in \mathbb{R}_D^+$ be a feasible packing associated to the set of items $\mathscr{I}$, which does not satisfy touching assumption. There is an item $i \in \mathscr{I}$ and $d \in \{1,\dots,D\}$ such that:

$$x_i^d \notin \{0\} \cup \left\{ x_j^d + w_j^d : j \in \mathscr{I} \setminus \{i\} \right\}.$$

Let $\tilde{x}_i^d = \max\{x_j^d + w_j^d : j \in \mathscr{I}, x_j^d + w_j^d \leq x_i^d\}$. And, for every $(d',i') \neq (d,i)$ ($d' \in \{1,\dots,D\}$, $i' \in \mathscr{I}$), let $\tilde{x}_{i'}^{d'} = x_{i'}^{d'}$. Let us check that $(\tilde{x}_j^1,\dots,\tilde{x}_j^D)_{j\in\mathscr{I}}$ is a feasible packing associated to $\mathscr{I}$.

We have, for every (d',i'), $\tilde{x}_{i'}^{d'} + w_{i'}^{d'} \leq x_{i'}^{d'} + w_{i'}^{d'} \leq W^{d'}$. Hence the constraint (1) is valid for these new coordinates.

It remains to check property (2). $\forall i' \in \mathscr{I} (i' \neq i)$, we have to prove that there is $d' \in \{1,\dots,D\}$ such that $[\tilde{x}_i^{d'}, \tilde{x}_i^{d'} + w_i^{d'}) \cap [\tilde{x}_{i'}^{d'}, \tilde{x}_{i'}^{d'} + w_{i'}^{d'}) = \emptyset$. But, for every $i' \in \mathscr{I} (i' \neq i)$, there is d' such that $[x_i^{d'}, x_i^{d'} + w_i^{d'}) \cap [x_{i'}^{d'}, x_{i'}^{d'} + w_{i'}^{d'}) = \emptyset$. So,

- If $d' \neq d$, then $[\tilde{x}_i^{d'}, \tilde{x}_i^{d'} + w_i^{d'}) \cap [\tilde{x}_{i'}^{d'}, \tilde{x}_{i'}^{d'} + w_{i'}^{d'}) = \emptyset$ because $\tilde{x}_i^{d'} = x_i^{d'}$.
- For $d' = d$, we have $x_{i'}^{d'} + w_{i'}^{d'} \leq x_i^{d'}$ or $x_i^{d'} + w_i^{d'} \leq x_{i'}^{d'}$.
  - If $x_{i'}^{d'} + w_{i'}^{d'} \leq x_i^{d'}$, then $\tilde{x}_{i'}^{d'} + w_{i'}^{d'} = x_{i'}^{d'} + w_{i'}^{d'} \leq \max\{x_j^{d'} + w_j^{d'} : j \in \mathscr{I}, x_j^{d'} + w_j^{d'} \leq x_i^{d'}\} = \tilde{x}_i^{d'}$.
  - If $x_i^{d'} + w_i^{d'} \leq x_{i'}^{d'}$, then $\tilde{x}_i^{d'} + w_i^{d} \leq x_i^{d'} + w_i^{d'} \leq x_{i'}^{d'} = \tilde{x}_{i'}^{d'}$.

Therefore, $[\tilde{x}_i^{d'}, \tilde{x}_i^{d'} + w_i^{d'}) \cap [\tilde{x}_{i'}^{d'}, \tilde{x}_{i'}^{d'} + w_{i'}^{d'}) = \emptyset$, for all $i' \in \mathscr{I} (i' \neq i)$. Furthermore, for every $i', i'' \in \mathscr{I} \setminus \{i\}$, we have $[\tilde{x}_{i'}^{d'}, \tilde{x}_{i'}^{d'} + w_{i'}^{d'}) \cap [\tilde{x}_{i''}^{d'}, \tilde{x}_{i''}^{d'} + w_{i''}^{d'}) = \emptyset$. Hence the constraint (2) is satisfied. Thus $(\tilde{x}_i^1,\dots,\tilde{x}_i^D)_{i\in\mathscr{I}}$ is a feasible packing associated to $\mathscr{I}$ with one item less not satisfying touching assumption than $(x_i^1,\dots,x_i^D)_{i\in\mathscr{I}}$. By iterating, we build this way a feasible packing associated to $\mathscr{I}$ satisfying touching assumption. □

118 Route de Narbonne, 31062 Toulouse Cedex 9, France
E-mail: arnaud.pecher@irit.fr

Let $G = (V,E)$ be a simple undirected graph. A stable set of G is a subset of V such that every pair of vertices are not adjacent. For every vertex $i \in V$, let $w_i \in \mathbb{R}$ be its weight. The weighted stability number $\alpha(G,\mathbf{w})$ of G is the maximal sum of weights of a stable set. Given a finite multi-set of intervals of $\mathbb{R}$, an interval graph $G = (V,E)$ is an undirected graph such that each interval corresponds to a vertex of the graph, and two vertices are adjacent if and only if the corresponding intervals overlap.

Following Fekete and Schepers [18, 19], a packing class is a collection of D graphs $G_1, \ldots, G_D$ with (shared) vertex set $\mathscr{I}$ and edge sets $E(G_d)$ such that:

P1: for every $d \in \{1,\ldots,D\}$, G_d is an interval graph,
P2: for every $d \in \{1,\ldots,D\}$, for every stable set S of G_d, $\sum_{s\in S} w_s^d \leq W^d$ (equivalently $\alpha(G_d,\mathbf{w^d}) \leq W^d$),
P3: $\bigcap_{d\in\{1,\ldots,D\}} E(G_d) = \emptyset$.

See Figure 2, for an illustration of the relationship between a feasible packing and a packing class.

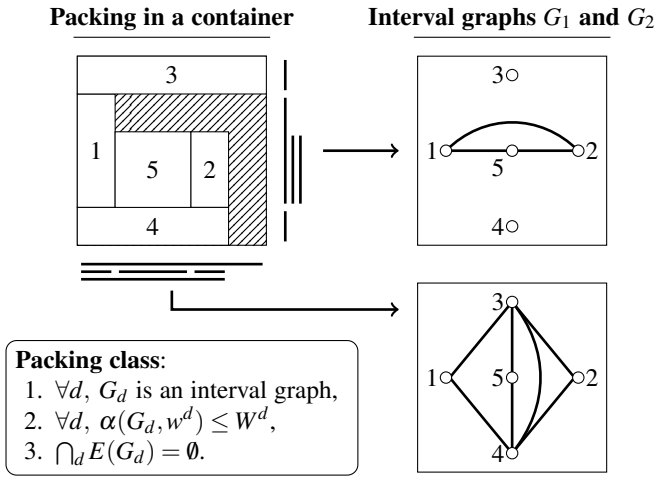


Fig. 2: A Fekete & Schepers' packing class with respect to a 2d-packing

Therefore to check whether a set of items is feasible, the algorithm of Fekete and Schepers enumerates all packing classes associated to $\mathscr{I}$. However, in some cases, there are distinct packings of a feasible set of items whose associated packing classes are different (see Figure 3). Hence there are still symmetry issues in this model. Furthermore Ferreira and Oliveira [20] noticed that some so-called degenerated packing classes are unnecessarily enumerated by Fekete and Schepers' algorithm.

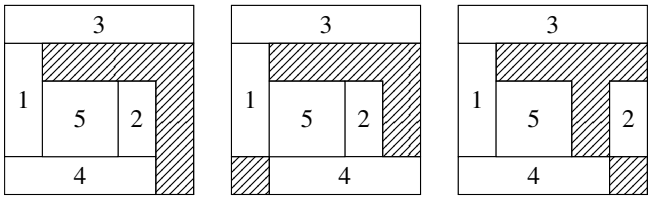


Fig. 3: Symmetry issues in Fekete & Schepers' model

A binary matrix $M \in \mathscr{M}_{n,m}(\mathbb{B})$ has the consecutive ones property if for every row i and $k \leq k'$, $M_{ik} = 1$ and $M_{ik'} = 1$ implies $M_{il} = 1$ for all $k \leq l \leq k'$ (for every row, the set of 1s occur consecutively).

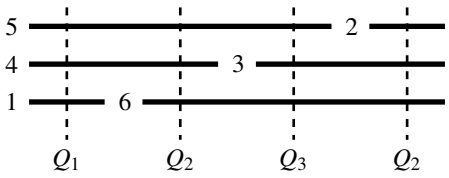
If $\mathbf{v}$ and $\mathbf{w}$ are two binary vectors of same size, we write $\mathbf{v} \underset{lex}{>} \mathbf{w}$, if $\mathbf{v}$ is of lexicographic order greater or equal than $\mathbf{w}$. Let M' and M'' be two matrices in $\mathscr{M}_{n,m}(\mathbb{B})$. $M' \underset{lex}{>} M''$ if $\exists l \in \{1,\ldots,m\}$ such that $\forall k < l, Q'_k = Q''_k$ and $Q'_l \underset{lex}{>} Q''_l$. We write $M' \sim M''$, if they have the same set of column vectors (M'' can be obtained from M' by permuting columns). In this case, M' and M'' are said to be equivalent.

Let G be a graph with n vertices ($\mathscr{V} = \{1,\dots,n\}$) and m maximal cliques $Q_1,\dots,Q_m$. A matrix $M \in \mathscr{M}_{n,m}(\mathbb{B})$ is a vertex/clique matrix of G if for every $1 \le i \le n$ and $1 \le k \le m$, the vertex i belongs to the maximal clique Q_k if and only if $M_{ik} = 1$ (see Example 3), that is each column is the characteristic vector of a maximal clique.

We are now ready to state Fulkerson and Gross' characterization of interval graphs:

Theorem 2 (Fulkerson and Gross (1965)) *A graph G is an interval graph if and only if there is a vertex/clique matrix of G which has the consecutive ones property.*

Example 3 Let G be the interval graph with the interval graph representation as depicted on the left part of the figure below. Then the matrix as defined on the right part is a vertex/clique matrix of G, which has the consecutive ones property.



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} Q_1 & Q_2 & Q_3 & Q_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

Without loss of generality, let $\mathscr{I}$ be a set of items and $\forall d \in \{1,\dots,D\}$, let $m^d \le n$. From now on, for every $d \in \{1,\dots,D\}$, let $M^d \in \mathscr{M}_{n,m^d}(\mathbb{B})$ be a matrix with the consecutive ones property. $\forall k \in \{1,\dots,m^d\}$, let $Q_k^d = \{i \in \mathscr{I} : M_{ik}^d = 1\}$ be the set of items in column k and let $C_k^d \in \mathbb{B}^n$ be the column k of matrix M^d (C_k^d is the characteristic vector of Q_k^d); a column C_k^d is said to be maximal if Q_k^d is not included in Q_l^d for every $l \neq k$. Let $Q_{m^d+1}^d = \emptyset$ and $\mathscr{Q}^d = \{Q_1^d,\dots,Q_{m^d}^d\}$. We define the width λ_k^d of Q_k^d by:

$$\lambda_k^d = \max_{\substack{i \in \mathscr{I} \\ i \in Q_k^d, i \notin Q_{k+1}^d}} \left\{ w_i^d - \sum_{1 \le l < k / i \in Q_l^d} \lambda_l^d \right\}. \tag{4}$$

A strip decomposition associated to $\mathscr{I}$ is a D-tuple of consecutive ones matrices $(M^1,\dots,M^D) \in \mathscr{M}(\mathbb{B})$ with non-zero rows such that:

– For every dimension $d \in \{1,\dots,D\}$,

$$\text{every column of } M^d \text{ is maximal,} \tag{5}$$

$$\forall i \in \mathscr{I}, \forall k \in \{1,\dots,m^d\} \text{ such that } i \in Q_k^d, \text{ we have } w_i^d - \sum_{1 \le l < k / i \in Q_l^d} \lambda_l^d > 0, \tag{6}$$

$$\sum_{k \in \{1,\dots,m^d\}} \lambda_k^d \le W^d. \tag{7}$$

– For every pair $i, j \in \mathscr{I} (i \neq j)$, there is a dimension $d \in \{1,\dots,D\}$ such that:

$$\forall k \in \{1,\dots,m^d\}, i \notin Q_k^d \text{ or } j \notin Q_k^d. \tag{8}$$

Figure 4 gives an illustration of a strip decomposition and of cliques widths.

We have the following basic Lemma, which is useful for some proofs:

Lemma 4 *For every $i \in \mathscr{I}$ and $d \in \{1,\dots,D\}$, $w_i^d \le \sum_{1 \le k \le m^d / i \in Q_k^d} \lambda_k^d$.*

Proof For every $i \in \mathscr{I}$ and $d \in \{1,\dots,D\}$, let $l \in \{1,\dots,m^d\}$ such that $i \in Q_l^d$ and $i \notin Q_{l+1}^d$.

We have $\lambda_l^d = \max_{\substack{j \in \mathscr{I} \\ j \in Q_l^d, j \notin Q_{l+1}^d}} \left\{ w_j^d - \sum_{1 \le k < l / j \in Q_k^d} \lambda_k^d \right\} \ge w_i^d - \sum_{1 \le k < l / i \in Q_k^d} \lambda_k^d$. Hence $w_i^d \le \sum_{1 \le k \le m^d / i \in Q_k^d} \lambda_k^d$. □

We now formally establish the link between feasible packings and strip decompositions (see Figure 4 again for an example).

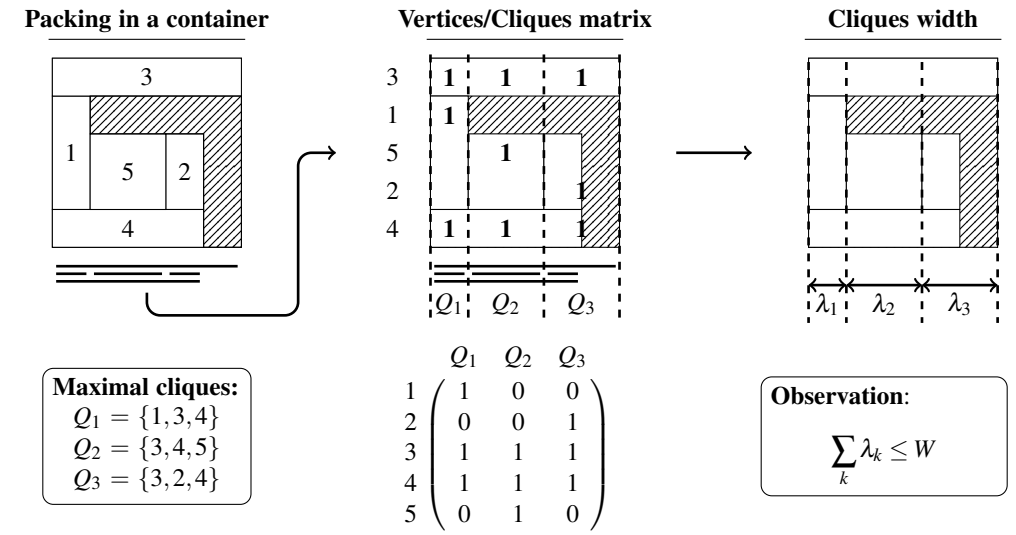


Fig. 4: A strip decomposition w.r.t. the first dimension of a 2d-orthogonal packing

Lemma 5 *If we have a feasible packing for $\mathscr{I}$ with touching assumption then there is a strip decomposition associated to $\mathscr{I}$.*

Proof By Lemma 1, there is an associated feasible packing $(x_i^1,\ldots,x_i^D) \in \mathbb{R}_D^+$ for every item $i \in \mathscr{I}$, which satisfies touching assumption. For all $d \in \{1,\ldots,D\}$, let $X^d = \{x \in \mathbb{R} : \exists i \in \mathscr{I}, x = x_i^d\}$. Let $m^d = |X^d|$. Write $X^d = \{X_1^d,\ldots,X_{m^d}^d\}$ with the convention $X_1^d < \cdots < X_{m^d}^d$. Since the feasible packing verifies touching assumption, for every $d \in \{1,\ldots,D\}, k \in \{2,\ldots,m^d\}$, there exists $i \in \mathscr{I}$ such that $X_k^d = x_i^d + w_i^d$. Moreover, $X_1^d = 0$. Let $X_{m^d+1}^d = \max\{x_i^d + w_i^d : i \in \mathscr{I}\}$.
Obviously, for every $d \in \{1,\ldots,D\}$, there is a surjective function $\sigma^d : \mathscr{I} \to \{1,\ldots,m^d\}$ such that $\forall i \in \mathscr{I}, x_i^d = X_{\sigma^d(i)}^d$.
$\forall d \in \{1,\ldots,D\}, k \in \{1,\ldots,m^d\}$, let $Q_k^d = \{i \in \mathscr{I} / x_i^d \leq X_k^d < x_i^d + w_i^d\}$. Let $M^d \in \mathscr{M}_{(n,m^d)}(\mathbb{B})$ be the boolean matrix such that $M_{ik}^d = 1$ if and only if $i \in Q_k^d$.

- M^d has the consecutive ones property. In fact, $\forall i \in \mathscr{I}$, let $k,l \in \{1,\ldots,m^d\}$, with $X_k^d \leq X_l^d$, such that $i \in Q_k^d$ and $i \in Q_l^d$. Then, by definition, $x_i^d \leq X_k^d < x_i^d + w_i^d$ and $x_i^d \leq X_l^d < x_i^d + w_i^d$. So, $\forall h \in \{1,\ldots,m^d\}$ such that $X_k^d \leq X_h^d \leq X_l^d$, we have this inequality: $x_i^d \leq X_k^d \leq X_h^d \leq X_l^d < x_i^d + w_i^d$. So, $i \in Q_h^d$. Thus M^d verifies the consecutive ones property.
- Let us check that every column of M^d is maximal. Let $1 \leq k < k' \leq m^d$ and C_k^d (resp. $C_{k'}^d$) be the characteristic vector of Q_k^d (resp. $Q_{k'}^d$). Since the feasible packing verifies touching assumption, let i (resp. i') $\in \mathscr{I}$ be an item such that $x_i^d + w_i^d = X_{k+1}^d$ (resp. $x_{i'}^d = X_{k'}^d$). By definition, $X_k^d < X_{k'}^d$. Obviously, $i' \notin Q_k^d$ and $i' \in Q_{k'}^d$. Hence, $Q_{k'}^d \not\subseteq Q_k^d$. Moreover, $X_{k+1}^d \leq X_{k'}^d$. Hence, $i \in Q_k^d$ and $i \notin Q_{k'}^d$. Thus, $Q_k^d \not\subseteq Q_{k'}^d$ and so C_k^d is maximal.

- $\forall d \in \{1,\ldots,D\}, \forall k \in \{1,\ldots,m^d\}$, let $\lambda_k^d = \max\limits_{\substack{i \in \mathscr{I} \\ i \in Q_k^d, i \notin Q_{k+1}^d}} \left\{ w_i^d - \sum\limits_{1 \leq l < k / i \in Q_l^d} \lambda_l^d \right\}$.
  Let $\mathscr{I}_k^d = \{i \in \mathscr{I} / x_i^d \leq X_k^d < x_i^d + w_i^d \leq X_{k+1}^d\} = \{i \in \mathscr{I} / i \in Q_k^d, i \notin Q_{k+1}^d\}$. As, the feasible packing verifies touching assumption, we have $X_k^d = \max\limits_{i \in \mathscr{I}_{k-1}^d} \{x_i^d + w_i^d\}$ for $k \in \{2,\ldots,m^d\}$.
  Let us show by induction that $\lambda_k^d = X_{k+1}^d - X_k^d > 0$, $\forall k \in \{1,\ldots,m^d\}$.
  . For $k = 1$, $\lambda_1^d = \max_{i \in \mathscr{I}_k^d} \left\{ w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d \right\} = \max_{i \in \mathscr{I}_k^d} \{w_i^d\} = X_2^d - X_1^d > 0$.

. For $k \geq 2$, $\lambda_k^d = \max_{i \in \mathscr{I}_k^d} \left\{ w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d \right\}$

$= \max_{i \in \mathscr{I}_k^d} \left\{ w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} (X_l^d - X_{l+1}^d) \right\}$ by induction hypothesis

$= \max_{i \in \mathscr{I}_k^d} \{ w_i^d + X_{\sigma^d(i)}^d - X_k^d \} = \max_{i \in \mathscr{I}_k^d} \{ x_i^d + w_i^d \} - X_k^d = X_{k+1}^d - X_k^d > 0.$

So, $\forall d \in \{1, \ldots, D\}$:

. $\sum_{k \in \{1, \ldots, m^d\}} \lambda_k^d = \sum_{k \in \{1, \ldots, m^d\}} (X_{k+1}^d - X_k^d) = X_{m^d+1}^d - X_1^d = \max\{x_i^d + w_i^d : i \in \mathscr{I}\} \leq W^d.$

. $\forall k \in \{1, \ldots, m^d\}, \forall i \in \mathscr{I}_k^d$, $w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d = w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} (X_l^d - X_{l+1}^d)$

$= w_i^d + X_{\sigma^d(i)}^d - X_k^d = x_i^d + w_i^d - X_k^d > 0.$

– Since we have a feasible packing, due to the constraint (2), we get $\forall i, j \in \mathscr{I} (i \neq j)$, $\exists d \in \{1, \ldots, D\} : [x_i^d, x_i^d + w_i^d) \cap [x_j^d, x_j^d + w_j^d) = \emptyset$. Hence, $x_i^d + w_i^d \leq x_j^d$ or $x_j^d + w_j^d \leq x_i^d$. Thus, $\forall k \in \{1, \ldots, m^d\}$, by definition of Q_k^d, $i \notin Q_k^d$ or $j \notin Q_k^d$.

Therefore $(M^1, \ldots, M^D)$ is a strip decomposition associated to $\mathscr{I}$. □

We now prove the converse:

Lemma 6 *If we have a strip decomposition associated to $\mathscr{I}$ then there is a feasible packing for $\mathscr{I}$ with touching assumption.*

Proof Let a D-tuple $(M^1, \ldots, M^d)$ of matrix which verify strip decomposition conditions. $\forall d \in \{1, \ldots, D\}$, $\forall i \in \mathscr{I}$, let f_i^d (resp. l_i^d) $\in \{1, \ldots, m^d\}$ be the smallest (resp. biggest) index $k \in \{1, \ldots, m^d\}$ such that $M_{ik}^d = 1$.
Let $x_i^d = \sum_{k \in \{1, \ldots, f_i^d - 1\}} \lambda_k^d$, $\forall d \in \{1, \ldots, D\}$, $\forall i \in \mathscr{I}$. We have to verify that the set of coordinates x_i^d defines a feasible packing.

– $\forall d \in \{1, \ldots, D\}, \forall i \in \mathscr{I}$, $x_i^d + w_i^d = \sum_{k \in \{1, \ldots, f_i^d - 1\}} \lambda_k^d + w_i^d \leq \sum_{k \in \{1, \ldots, f_i^d - 1\}} \lambda_k^d + \sum_{k \in \{f_i^d, \ldots, l_i^d\}} \lambda_k^d \leq W^d$.
Hence the constraint (1) is satisfied.
– $\forall i, j \in \mathscr{I} (i \neq j)$, it exists $d \in \{1, \ldots, D\}$ such that $\forall k \in \{1, \ldots, m^d\}$, $i \notin Q_k^d$ or $j \notin Q_k^d$. As, M^d is a consecutive ones matrix, then, $l_i^d < f_j^d$ or $l_j^d < f_i^d$. So, $x_i^d + w_i^d \leq x_j^d$ or $x_j^d + w_j^d \leq x_i^d$. Thus the constraint (2) is satisfied.
– It remains to check that the touching assumption holds. $\forall d \in \{1, \ldots, D\}, i \in \mathscr{I}$, $x_i^d = \sum_{k \in \{1, \ldots, f_i^d - 1\}} \lambda_k^d$. Let $j = \underset{\substack{i' \in \mathscr{I} \\ i' \in Q_{f_i^d - 1}^d, i' \notin Q_{f_i^d}^d}}{\operatorname{argmax}} \left\{ w_{i'}^d - \sum_{1 \leq l < f_i^d - 1 / i' \in Q_l^d} \lambda_l^d \right\}$.
So, $\lambda_{f_i^d - 1}^d = w_j^d - \sum_{1 \leq l < f_i^d - 1 / j \in Q_l^d} \lambda_l^d$. Thus, $w_j^d = \sum_{f_j^d \leq l < f_i^d - 1} \lambda_l^d$. So, $x_i^d = \sum_{k \in \{1, \ldots, f_j^d - 1\}} \lambda_k^d + \sum_{k \in \{f_j^d, \ldots, f_i^d - 1\}} \lambda_k^d = x_j^d + w_j^d$.
□

2.2 The algorithm to check feasibility

Due to Lemmas 5 and 6, to check feasibility, we only have to design an algorithm which returns whether there is a strip decomposition associated to the set of items $\mathscr{I}$. We proceed dimension by dimension, by enumerating all consecutive ones matrices which satisfy the strip decomposition constraints (5), (6), (7) and (8).

The core of the recursion to enumerate consecutive ones matrices is described in Algorithm 1. Notice that the number of columns of a matrix of a strip decomposition is at most n (the number of items), but we do not known it when we are looking for a strip decomposition. Therefore our algorithm enumerates square $n \times n$-matrices, such that there is an index m implying all columns with index at most m are maximal, and the remaining ones are 0 columns.

By applying algorithm 1, we consider at this stage only one dimension, and this is a main point of our approach: almost all the work is done in first dimension.

To ensure that constraint (8), is satisfied, we proceed like this: for every dimension $d \in \{1,\dots,D-1\}$, let S_d denote the set of all matrices output by Algorithm 1 w.r.t. dimension d. Then for the last dimension $d = D$, for every $(D-1)$-tuple of matrices $(M^1,\dots,M^{D-1}) \in S_1 \times \dots \times S_{D-1}$, we look for a consecutive ones matrix by using Algorithm 1, such that constraint (8) is satisfied. If any, we have a strip decomposition, and the set of items is feasible. If we fail to find out such a tuple of matrices, the set of items is not feasible.

2.3 Early detection of unfeasibility

Let $\mathscr{I}$ be a feasible set of items. In this subsection, we exhibit additional constraints which are valid for all strip decompositions associated to $\mathscr{I}$. These constraints are used in the algorithm to reduce the enumeration tree of strip decompositions.

Available width for items which are not packed yet Assume that the first k columns in Algorithm 1 are set. Every item which has only 0 entries in these columns has to be packed in the remaining columns. Therefore, the following constraint is a valid constraint (see Figure 5).

Lemma 7 *For every $d \in \{1,\dots,D\}$ and $k \in \{1,\dots,m^d\}$,*

$$\sum_{l \in \{1,\dots,k\}} \lambda_l^d + \max_{i \notin \{Q_1^d,\dots,Q_k^d\}} w_i^d \leq W^d. \tag{9}$$

Proof Indeed, $\forall d \in \{1,\dots,D\}$ and $k \in \{1,\dots,m^d\}$, $\max_{i \notin \{Q_1^d,\dots,Q_k^d\}} w_i^d \leq \sum_{l \in \{k+1,\dots,m^d\}} \lambda_l^d$. □

"Height" of the column In the bi-dimensional case, a maximal clique Q w.r.t. one dimension induces a stable set in the other dimension. Therefore, for every column Q, the sum of weights of the items of Q (the "height" of Q) can not exceed the size of the container in the other dimension. In D dimensions, we have this valid following constraint:

Lemma 8 *For every $d \in \{1,\dots,D\}$ and $k \in \{1,\dots,m^d\}$,*

$$\sum_{i \in Q_k^d} \left(\prod_{\substack{d' \in \{1,\dots,D\} \\ d' \neq d}} w_i^{d'} \right) \leq \prod_{\substack{d' \in \{1,\dots,D\} \\ d' \neq d}} W^{d'}. \tag{10}$$

Require: (int) `n`, (bool[n][n]) `mat`
Ensure: print all $n \times n$ matrices with consecutive ones property, satisfying strip decompositions constraints (5), (6), and (7), and no 0 row

```
   recurse(int row, int column)
 1: if column > n+1 then
 2:     return
 3: end if
 4: if row > n+1 then
 5:     if ((every column of index at most column is maximal) and
        (constraints (6) and (7) are satisfied)) then
 6:         if number of non empty rows = n then
 7:             print matrix mat
 8:         else
 9:             recurse(1,column+1)
10:         end if
11:     end if
12:     return
13: end if
14: mat(row,column)=true
15: if mat is still a consecutive ones matrix then
16:     recurse(row+1,column)
17: end if
18: mat(row,column)=false
19: recurse(row+1,column)
```

Algorithm 1: recursive generation of consecutive ones matrices satisfying strip decompositions constraints (5), (6), and (7) without 0 rows

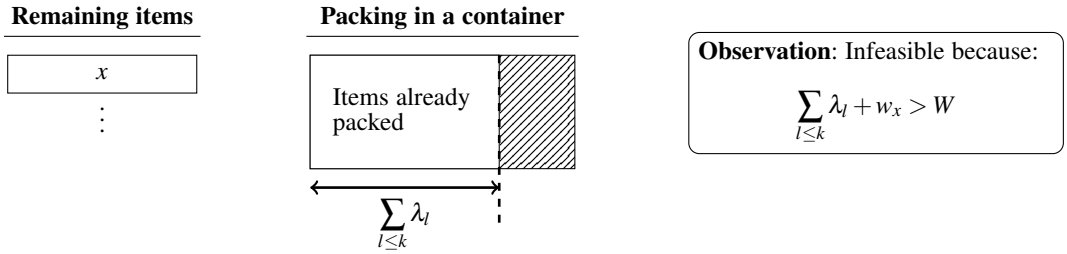


Fig. 5: Width of item x has to fit in the grey part of the container

Proof Let $d \in \{1,\dots,D\}$ and $k \in \{1,\dots,m^d\}$. Due to Lemma 4,

$$\sum_{i\in Q_k^d}\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}} w_i^{d'}\right) \leq \sum_{i\in Q_k^d}\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}\sum_{\substack{l\in\{1,\dots,m^{d'}\}\\ i\in Q_l^{d'}}} \lambda_l^{d'}\right).$$

Let $\mathscr{Z} = \{1,\dots,m^1\}\times\cdots\times\{1,\dots,m^{d-1}\}\times\{1,\dots,m^{d+1}\}\times\cdots\times\{1,\dots,m^D\}$. For every $i\in\mathscr{I}$ and $d'\in\{1,\dots,D\}$ with $(d'\neq d)$, let $L_i^{d'} = \left\{l\in\{1,\dots,m^{d'}\} : i\in Q_l^{d'}\right\}$. Let $\mathscr{Z}_i = L_i^1\times\cdots\times L_i^{d-1}\times L_i^{d+1}\times\cdots\times L_i^D \subseteq \mathscr{Z}$.
Notice that $\mathscr{Z}_i$ is not empty since $L_i^{d'}\neq\emptyset$ for every $d'\in\{1,\dots,D\}$ with $d'\neq d$.
Let P, Q and, for every $i\in\mathscr{I}$, P_i be the multivariate polynomials of $\mathbb{R}[X_{z\in\mathscr{Z}}]$ defined by:

$$\begin{aligned}
Q &= \sum_{(l_1,\dots,l_{d-1},l_{d+1},\dots,l_D)\in\mathscr{Z}} X_{(1,l_1)}X_{(2,l_2)}\dots X_{(d-1,l_{d-1})}X_{(d+1,l_{d+1})}\dots X_{(D,l_D)},\\
P_i &= \sum_{(l_1,\dots,l_{d-1},l_{d+1},\dots,l_D)\in\mathscr{Z}_i} X_{(1,l_1)}X_{(2,l_2)}\dots X_{(d-1,l_{d-1})}X_{(d+1,l_{d+1})}\dots X_{(D,l_D)},\\
P &= \sum_{i\in Q_d^k} P_i.
\end{aligned}$$

For every $d'\in\{1,\dots,D\}$ with $(d'\neq d)$, $\forall(l_1,\dots,l_{d-1},l_{d+1},\dots,l_D)\in\mathscr{Z}$, let the variable $X_{(d',l_{d'})}$ take the value $\lambda_{l_{d'}}^{d'}$ and denote by $P(\lambda)$ (resp. $Q(\lambda)$, $P_i(\lambda)$) the corresponding value of P (resp. Q, P_i for every $i\in\mathscr{I}$). By definition, $\forall i,j\in\mathscr{I}(i\neq j)$, $\exists d'\in\{1,\dots,D\}$ such that $\forall l\in\{1,\dots,m^{d'}\}$, $i\notin Q_l^{d'}$ or $j\notin Q_l^{d'}$. Hence, for every pair of distinct elements $i,j\in Q_k^d$, we have $\mathscr{Z}_i\cap\mathscr{Z}_j=\emptyset$. Hence, the polynomial P is an homogeneous polynomial of degree $D-1$ such that all non-zero coefficients are equal to 1.

$$\sum_{i\in Q_k^d}\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}\sum_{\substack{l\in\{1,\dots,m^{d'}\}\\ i\in Q_l^{d'}}}\lambda_l^{d'}\right) = \sum_{i\in Q_d^k} P_i(\lambda) = P(\lambda) \leq Q(\lambda) = \prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}\left(\sum_{l\in\{1,\dots,m^{d'}\}}\lambda_l^{d'}\right)$$

Therefore, the inequality (10) is a valid inequality as for all $d'\in\{1,\dots,D\}$, $\sum_{l\in\{1,\dots,m^{d'}\}}\lambda_l^{d'} \leq W^{d'}$. □

Available volume for items which are not packed yet Assume that the first k columns in Algorithm 1 are set. The sum of all items volume which have only 0 entries in these columns has to be packed in the remaining volume. Therefore, the following constraint is a valid constraint (see Figure 6).

Lemma 9 *For every $d\in\{1,\dots,D\}$ and $k\in\{1,\dots,m^d\}$,*

$$\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}} W^{d'}\right)\left(\sum_{l\in\{1,\dots,k\}}\lambda_l^{d'}\right) + \sum_{i\in Q_k^d\cap Q_{k+1}^d}\left(\left(w_i^d - \sum_{\substack{l\in\{1,\dots,k\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}} w_i^{d'}\right)\right) +$$

$$\sum_{i\notin\{Q_1^d,\dots,Q_k^d\}}\left(\prod_{d'\in\{1,\dots,D\}} w_i^{d'}\right) \leq \prod_{d'\in\{1,\dots,D\}} W^{d'} \tag{11}$$

Proof Due to Lemma 4, $\forall d\in\{1,\dots,D\}$ and $k\in\{1,\dots,m^d\}$, we have:

$$-\sum_{i\in Q_k^d\cap Q_{k+1}^d}\left(\left(w_i^d-\sum_{\substack{l\in\{1,\dots,k\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)\leq\sum_{i\in Q_k^d\cap Q_{k+1}^d}\left(\left(\sum_{\substack{l\in\{k+1,\dots,m^d\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)$$

$$-\sum_{i\notin\{Q_1^d,\dots,Q_k^d\}}\left(\prod_{d'\in\{1,\dots,D\}}w_i^{d'}\right)\leq\sum_{i\notin\{Q_1^d,\dots,Q_k^d\}}\left(\left(\sum_{\substack{l\in\{k+1,\dots,m^d\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)$$

Hence,

$$\sum_{i\in Q_k^d\cap Q_{k+1}^d}\left(\left(w_i^d-\sum_{\substack{l\in\{1,\dots,k\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)+\sum_{i\notin\{Q_1^d,\dots,Q_k^d\}}\left(\prod_{d'\in\{1,\dots,D\}}w_i^{d'}\right)\leq$$

$$\sum_{i\in\{Q_{k+1}^d,\dots,Q_{m^d}^d\}}\left(\left(\sum_{\substack{l\in\{k+1,\dots,m^d\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)=\sum_{l\in\{k+1,\dots,m^d\}}\left(\lambda_l^d\left(\sum_{i\in Q_l^d}\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)$$

Due to the valid inequality (10), we get

$$\sum_{i\in Q_k^d\cap Q_{k+1}^d}\left(\left(w_i^d-\sum_{\substack{l\in\{1,\dots,k\}\\ i\in Q_l^d}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}w_i^{d'}\right)\right)+\sum_{i\notin\{Q_1^d,\dots,Q_k^d\}}\left(\prod_{d'\in\{1,\dots,D\}}w_i^{d'}\right)\leq$$

$$\left(\sum_{l\in\{k+1,\dots,m^d\}}\lambda_l^d\right)\left(\prod_{\substack{d'\in\{1,\dots,D\}\\ d'\neq d}}W^{d'}\right)$$

Therefore, the inequality (11) is a valid inequality. □

So, due to lemmas 7, 8, 9, we only have to generate columns satisfying constraints (9), (10) and (11).

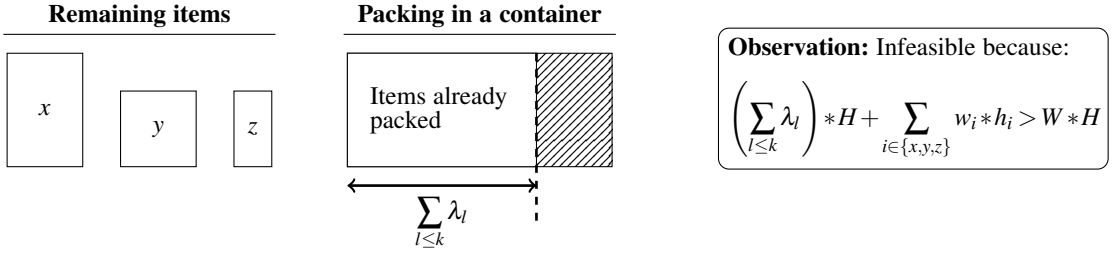


Fig. 6: Sum of areas of remaining items to be packed can not exceed the area of the grey part of the container

2.4 Breaking some symmetry issues

Notice that the first matrix printed by Algorithm 1 is the biggest matrix satisfying strip decomposition constraints without 0 rows, with respect to the lexicographic order (as defined in Section 1). Hence, if during the recursion, we are able to detect that the packing is feasible, then a solution with biggest lexicographic order should have been already found, in which case we can safely stop it and return that the packing is infeasible.

Let us see a small example: assume that in the enumerating process, we are in step 1 below (the star (resp. $-$) denotes current position (resp. unset positions))

$$\underset{\text{step 1}}{\begin{pmatrix}1&0&0&-\\0&1&0&-\\0&0&*&-\\0&0&-&-\end{pmatrix}}\quad\Bigg|\quad\underset{\text{step 2}}{\begin{pmatrix}0&1&0&-\\1&0&0&-\\0&0&*&-\\0&0&-&-\end{pmatrix}}$$

Then step 2 must have been visited before by the recursion, hence a feasible solution, if any, should have been found before: in this case, there are no feasible solutions.

By detecting such configurations, this is how we handle symmetry issues. This contributed mainly to our algorithm's efficiency. This subsection is devoted to such breaking symmetries procedures.

Such symmetry issues arise naturally when two items are identical (they have same widths w.r.t. all dimensions), since the corresponding rows may be swapped safely. These symmetry issues are easily dealt with.

Let $M \in \mathscr{M}_{n,m}(\mathbb{B})$ be a 0/1 matrix. For every $1 \leq p \leq m$ and $1 \leq q \leq m-p$, we denote by $\mathrm{SM}_{p,q}(M)$ the submatrix M' of M made of the columns numbered from p to $p+q-1$, that is for every $1 \leq i \leq n$ and $1 \leq j \leq q$, $M'_{i,j} = M_{i,j+p-1}$.

A submatrix $\mathrm{SM}_{p,q}(M)$ is a *block* of M if for every i, j such that $M_{i,j} = 1$ and $p \leq j \leq p+q-1$, we have $M_{i,1} = M_{i,2} = \ldots, M_{i,p-1} = 0$ and $M_{i,p+q} = M_{i,p+q+1} = \ldots = M_{i,m} = 0$. We denote by $\mathrm{B}_{p,q}(M)$ a submatrix $\mathrm{SM}_{p,q}(M)$ of M which is a block of M (see Figure 7). Roughly speaking with respect to strip decompositions, every item which belongs to a clique of a block can not belong to the cliques outside of the block.

$$\begin{pmatrix} 0 & \boxed{\begin{matrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{matrix}} & 0 \\ 1 & & 0 \\ 0 & & 0 \\ 0 & & 0 \\ 0 & & 1 \\ 1 & & 0 \end{pmatrix}$$
block

Fig. 7: Block $\mathrm{B}_{2,2}(M)$

Lemma 10 *Let $(M^1, \ldots, M^D)$ be a strip decomposition. For every $d \in \{1, \ldots, D\}$ such that M^d has two blocks $B_{p_1,q_1}(M^d)$ and $B_{p_2,q_2}(M^d)$ with $p_1 + q_1 \leq p_2$ and $B_{p_2,q_2}(M^d) \underset{lex}{>} B_{p_1,q_1}(M^d)$ then there is a matrix M'^d such that $M'^d \underset{lex}{>} M^d$ and the D-tuple of matrices $(M^1, \ldots, M'^d, \ldots, M^D)$ is an equivalent strip decomposition (see Figure 8).*

Proof Let M'^d be the matrix obtained from M^d by swapping the two blocks $B_{p_1,q_1}(M^d)$ and $B_{p_2,q_2}(M^d)$: the columns C'^d_k of M'^d are given by

$$C'^d_k = \begin{cases} C^d_k & \text{if } k < p_1 \\ C^d_{k+p_2-p_1} & \text{if } p_1 \leq k < p_1+q_2 \\ C^d_{k+q_1-q_2} & \text{if } p_1+q_2 \leq k < p_2+q_2-q_1 \\ C^d_{k+p_1+q_1-p_2-q_2} & \text{if } p_2+q_2-q_1 \leq k < p_2+q_2 \\ C^d_k & \text{if } p_2+q_2 \leq k \end{cases}.$$

It is straightforward to check that $(M^1, \ldots, M'^d, \ldots, M^D)$ is an equivalent strip decomposition to $(M^1, \ldots, M^d, \ldots, M^D)$ such that $M'^d \underset{lex}{>} M^d$. □

Hence, due to Lemma 10, we stop the main recursion when there are two blocks which are not in decreasing lexicographic order (right case in Figure 8).

Notice that given a block, it is possible to permute the columns such that the first column is bigger w.r.t. the lexicographic order than the last one (see Figure 9). We are going to push this argument further to a slightly more general structure than blocks.

A submatrix $\mathrm{SM}_{p,q}(M)$ is an *inner block* of M if, for every i, j, such that $M_{i,j} = 1$ and $p \leq j \leq p+q-1$, we have $(M_{i,1} = M_{i,2} = \ldots = M_{i,p-1} = 0$ and $M_{i,p+q} = M_{i,p+q+1} = \ldots = M_{i,m} = 0)$ or $(M_{i,p} = M_{i,p+1}, \ldots = M_{i,p+q-1} = 1)$. We denote by $\mathrm{IB}_{p,q}(M)$ a submatrix $\mathrm{SM}_{p,q}(M)$ of M which is an inner block of M (see Figure 10). Roughly speaking with respect to strip decompositions, every item which belongs to a clique of an inner block must belong to all cliques of the inner block or can not belong to the cliques outside of the inner block. An inner block $IB_{p,q}(M)$ is *minimal* if $Q_p \underset{lex}{>} Q_{p+q-1}$.

Lemma 11 *Let $(M^1, \ldots, M^D)$ be a strip decomposition associated to the set of items $\mathscr{I}$. For every $d \in \{1, \ldots, D\}$ such that M^d an inner block $IB_{p,q}(M^d)$ not "minimal" then there is a matrix M'^d such that $M'^d \underset{lex}{>} M^d$ and the D-tuple of matrices $(M^1, \ldots, M'^d, \ldots, M^D)$ is another strip decomposition associated to $\mathscr{I}$ (see Figure 11).*

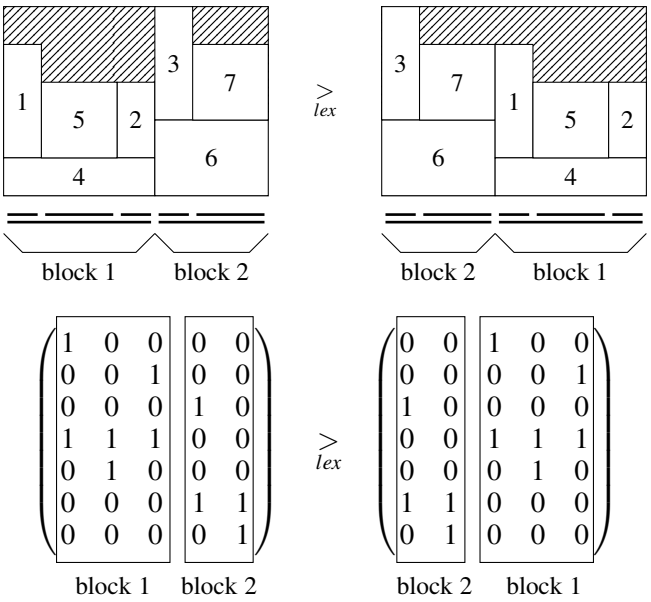


Fig. 8: Permuting two blocks gives two equivalent strip decompositions

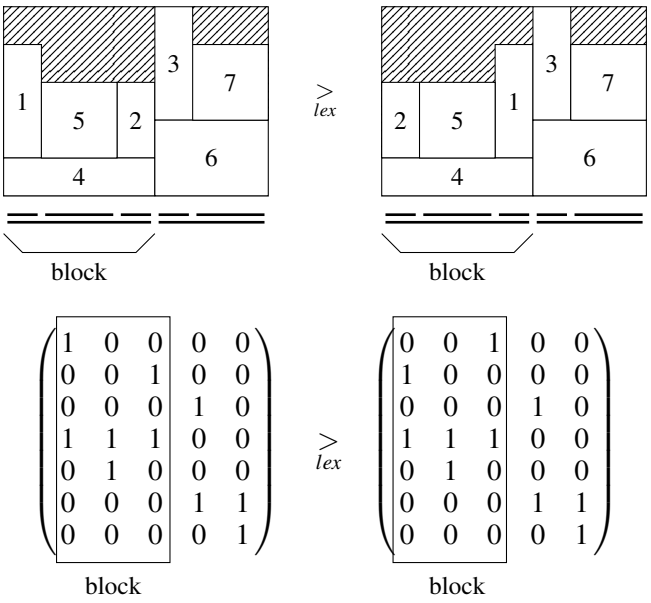


Fig. 9: Permutation of columns within a block

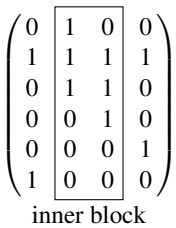


Fig. 10: Inner block $IB_{2,2}$

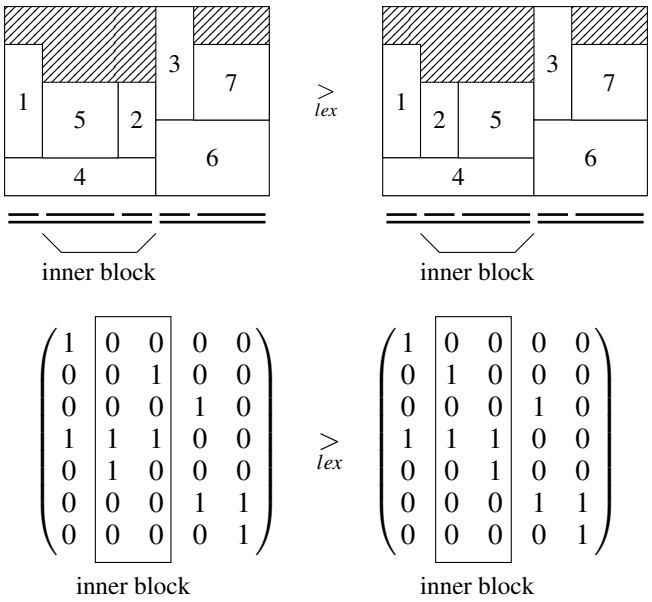


Fig. 11: Permutation of columns within an inner block

Proof Let $d \in \{1,\dots,D\}$ such that M^d has an inner block matrix $IB_{p,q}(M^d)$ which is not minimal.
Let $\alpha^d_{p,q} = \sum_{k\in\{1,\dots,p-1\}} \lambda^d_k$ and $\beta^d_{p,q} = \sum_{k\in\{1\dots,p+q-1\}} \lambda^d_k$. For every item $i \in \mathscr{I}$ and for every $\delta \in \{1,\dots,D\}$, let f^δ_i (resp. l^δ_i) $\in \{1,\dots,m^\delta\}$ be the smallest (resp. biggest) index $k \in \{1,\dots,m^\delta\}$ such that $M^\delta_{i,k} = 1$; and let $x^\delta_i = \sum_{k\in\{1,\dots,f^\delta_i-1\}} \lambda^\delta_k$.

According to the proof of Lemma 6, $(x^1_i,\dots,x^D_i)_{i\in\mathscr{I}}$ is a feasible packing. We are going to build another feasible packing $(x'^1_i,\dots,x'^D_i)_{i\in\mathscr{I}}$ with $x'^\delta_i = x^\delta_i$ whenever $\delta \in \{1,\dots,D\}$ $(\delta \neq d)$ and $i \in \mathscr{I}$ such that the associated strip decomposition, as defined in the proof of Lemma 5, has bigger lexicographic order, with respect to dimension d:

- $\forall i \in \mathscr{I}$ and $\delta \in \{1,\dots,D\}$ with $\delta \neq d$, let $x'^\delta_i = x^\delta_i$.
- $\forall i \in \mathscr{I}$ such that $f^d_i < p$ or $l^d_i > p+q-1$, let $x'^d_i = x^d_i = \sum_{k\in\{1,\dots,f_i-1\}} \lambda^d_k$.
- $\forall i \in \mathscr{I}$ such that $f^d_i \geq p$ and $l^d_i \leq p+q-1$, let $x'^d_i = \alpha^d_{p,q} + (\beta^d_{p,q} - \sum_{k\in\{1,\dots,l^d_i\}} \lambda^d_k)$. Hence, we have:

$$x'^d_i = \sum_{k\in\{1,\dots,p-1\}} \lambda^d_k + \sum_{k\in\{l^d_i+1,\dots,p+q-1\}} \lambda^d_k.$$

Let us check that inequalities (1) with respect to dimension d of a feasible packing are satisfied :

- $\forall i \in \mathscr{I}$ such that $f^d_i < p$ or $l^d_i > p+q-1$, $x'^d_i + w^d_i \leq W^d$.
- $\forall i \in \mathscr{I}$ such that $f^d_i \geq p$ and $l^d_i \leq p+q-1$, due to Lemma (4), we have:

$$x'^d_i + w^d_i \leq \sum_{k\in\{1,\dots,p-1\}} \lambda^d_k + \sum_{k\in\{l^d_i+1,\dots,p+q-1\}} \lambda^d_k + \sum_{k\in\{f^d_i,\dots,l^d_i\}} \lambda^d_k \leq W^d.$$

Let us check that constraints (2) of a feasible packing are satisfied. If $(i,j) \in \mathscr{I}^2$ $(i \neq j)$ is a pair of items such that $[x^d_i, x^d_i + w^d_i) \cap [x^d_j, x^d_j + w^d_j) \neq \emptyset$, then there is nothing to prove. Assume that $(i,j) \in \mathscr{I}^2$ $(i \neq j)$ is a pair of items such that $[x^d_i, x^d_i + w^d_i) \cap [x^d_j, x^d_j + w^d_j) = \emptyset$. We are going to show that $[x'^d_i, x'^d_i + w^d_i) \cap [x'^d_j, x'^d_j + w^d_j) = \emptyset$ (which implies that constraint (2) is satisfied for the pair (i,j)). There are four cases:

- If $(f^d_i < p$ or $l^d_i > p+q-1)$ and $(f^d_j < p$ or $l^d_j > p+q-1)$ then $x'^d_i = x^d_i$ and $x'^d_j = x^d_j$. Thus $[x'^d_i, x'^d_i + w^d_i) \cap [x'^d_j, x'^d_j + w^d_j) = \emptyset$.
- If $(f^d_i \geq p$, $l^d_i \leq p+q-1)$ and $(f^d_j < p)$ then $x^d_j + w^d_j \leq x^d_i$. By definition of an inner block, we have $l^d_j < p$. Thus, due to Lemma (4), we have:

$$x'^d_j + w^d_j = x^d_j + w^d_j \leq \sum_{k\in\{1,\dots,f^d_j-1\}} \lambda^d_k + \sum_{k\in\{f^d_j,\dots,l^d_j\}} \lambda^d_k \leq \sum_{k\in\{1,\dots,p-1\}} \lambda^d_k \leq x'^d_i.$$

– If ($f_i^d \geq p$, $l_i^d \leq p+q-1$) and ($l_j^d > p+q-1$) then $x_j^d \geq x_i^d + w_i^d$. By definition of an inner block, we have $f_j^d > p+q-1$. Thus, due to Lemma (4), we have:

$$x_j'^d = x_j^d \geq \sum_{k\in\{1,\ldots,f_j^d-1\}} \lambda_k^d \geq \sum_{k\in\{1,\ldots,p+q-1\}} \lambda_k^d \geq x_i'^d + w_i^d.$$

– If ($f_i^d \geq p$, $l_i^d \leq p+q-1$) and ($f_j^d \geq p$, $l_j^d \leq p+q-1$). We suppose $x_j^d + w_j^d \leq x_i^d$, then $l_j^d < f_i^d$. Due to Lemma (4), we have:

$$\begin{aligned} x_i'^d + w_i^d &\leq \sum_{k\in\{1,\ldots,p-1\}} \lambda_k^d + \sum_{k\in\{l_i^d+1,\ldots,p+q-1\}} \lambda_k^d + \sum_{k\in\{f_i^d,\ldots,l_i^d\}} \lambda_k^d \\ &\leq \sum_{k\in\{1,\ldots,p-1\}} \lambda_k^d + \sum_{k\in\{l_j^d+1,\ldots,p+q-1\}} \lambda_k^d \leq x_j'^d. \end{aligned}$$

Therefore, $(x_i'^1,\ldots,x_i'^D)_{i\in\mathscr{I}}$ is a feasible packing. From Lemma 1, we get a feasible packing $(x_i"^1,\ldots,x_i"^D)_{i\in\mathscr{I}}$ with the touching assumption, such that $x_i^\delta = x"_i^\delta$ for all $i \in \mathscr{I}$, whenever $\delta \neq d$. As defined in the proof of Lemma 5, we have an associated strip decomposition $(M^1,\ldots,M"^d,\ldots,M^D)$. The first $p-1$ columns of $M"^d$ are identical to M^d (as the coordinates of the first items are identical). By hypothesis, $Q^d_{p+q-1} \underset{lex}{\geq} Q^d_p$. Let a (resp. b) be the smallest item such that $M^d_{a,p} = 1$ and $M^d_{a,p+q-1} = 0$ (resp. $M^d_{b,p} = 0$ and $M^d_{b,p+q-1} = 1$). Due to the definition of inner block, $l_b^d = p+q-1$ and so $x"_b^d = \sum_{k\in\{1,\ldots,p-1\}} \lambda_k^d = x_a^d$. According to the touching assumption, there is $j \in \mathscr{I}$ such that $x_a^d = x_j^d + w_j^d$. So, $x"_b^d = x"_j^d + w_j^d$ and thus the item b satisfies touching assumption. Hence $b \in Q"_p^d$. Thus, $Q"_p^d \underset{lex}{\geq} Q_p^d$ and $M"^d \underset{lex}{\geq} M^d$. □

Hence, due to Lemma 11, we stop the main recursion when there is an inner block which is not minimal (right case in Figure 11).

3 The two-dimensional orthogonal knapsack problem

The main benchmarks are devoted to the bi-dimensional orthogonal knapsack problem. Our algorithm only checks if a given set of items admits a feasible packing. Therefore, to run these benchmarks, we used a branch-and-bound algorithm (see subsection 3.1), calling our algorithm to check feasibility.

3.1 Knapsack algorithm

To select a subset of items $\mathscr{I}$ among all items, we use a classical branch-and-bound enumeration of the knapsack problem where the knapsack constraint is that the overall area of the selected items is less or equal than the container's area. For a given selected subset of items, we apply some dual feasible functions in order to try to early detect unfeasibility: a function $f : [0,1] \to [0,1]$ is *dual feasible* (DFF) [17] if for every finite set S of positive real numbers, we have $\sum_{x\in S} x \leq 1 \Rightarrow \sum_{x\in S} f(x) \leq 1$.

Carlier and Neron introduced discrete dual feasible functions [9], as follows: let $C \in EnsR^+$, a *discrete dual feasible function* f_C is a function $f_C : [0,C] \to [0,C']$ (where $C' = f_C(C)$) such that for every finite set S of positive real numbers $\forall S, \quad \sum_{x\in S} x \leq C \Rightarrow \sum_{x\in S} f_C(x) \leq f_C(C) = C'$. They proved the following necessary condition for a subset of items to be feasible.

Theorem 12 *A subset of items I is feasible, only if for every D-tuple of discrete feasible function $(f^1_{W_1},\ldots,f^D_{W_D})$, we have*

$$\sum_{i\in I} \left(\prod_{d=1}^{D} f^d(w_i^d) \right) \leq \prod_{d=1}^{D} f^d(W^d) \tag{12}$$

We apply Theorem 12 with the following discrete dual feasible functions ($k = 2,3,4$), for every dimension d:

$f^d_{k,FS,1} : [0,W^d] \to [0,kW^d]$

$$x \mapsto \begin{cases} xk, & \text{if } \frac{x(k+1)}{W^d} \in \mathbb{N}, \\ \left\lfloor \frac{x(k+1)}{W^d} \right\rfloor C, & \text{otherwise.} \end{cases}$$

$f^d_{k,CCM,1} : [0,W^d] \to [0,2\left\lfloor \frac{W^d}{k} \right\rfloor]$

$$x \mapsto \begin{cases} 2\left(\lfloor \frac{W^d}{k} \rfloor - \lfloor \frac{W^d-x}{k} \rfloor\right), & \text{if } x > \frac{W^d}{2}, \\ \lfloor \frac{W^d}{k} \rfloor, & \text{if } x = \frac{W^d}{2}, \\ \lfloor \frac{x}{k} \rfloor, & \text{if } x < \frac{W^d}{2}. \end{cases}$$

To solve the knapsack problem, we may use a classical solver such as IBM Ilog CPLEX, FICO Xpress or GLPK. This would not improve significantly the computional time as the main issue is to check the feasibility of the packing. Hence we implemented our own knapsack algorithm.

3.2 Computational results on standard benchmarks

Benchmark	**bin size**	**nb of items in opt.**	**opt. sol.**	**JP** OKP nodes	**JP** OPP calls	**FS** OKP nodes	**FS** OPP calls
ngcut1	10×10	5	164	23	13	19	1
ngcut2	10×10	5	230	74	19	5	0
ngcut3	10×10	7	247	29	19	25	6
ngcut4	15×10	6	268	7	6	1	0
ngcut5	15×10	6	358	8	7	1	0
ngcut6	15×10	7	289	38	26	15	5
ngcut7	20×20	8	430	8	8	0	0
ngcut8	20×20	8	834	67	62	53	23
ngcut9	20×20	11	924	21	18	3	0
ngcut10	30×30	6	1452	7	0	1	0
ngcut11	30×30	9	1688	115	78	36	10
ngcut12	30×30	9	1865	210	46	48	14
gcut1	250×250	3	48368	69	18	33	0
gcut2	250×250	6	59798	2873	169	519	51
gcut3	250×250	6	61275	16503	564	2234	235
gcut4	250×250	4	61380	310308	36026	72159	18316
gcut5	500×500	5	195582	92	45	52	13
gcut6	500×500	4	236305	1348	78	278	22
gcut7	500×500	4	240143	8471	207	852	124
gcut8	500×500	4	245758	313847	14685	55485	9037
gcut9	1000×1000	5	939600	75	10	12	2
gcut10	1000×1000	5	937349	1176	88	335	31
gcut11	1000×1000	6	969709	29446	1426	1616	212
gcut12	1000×1000	5	979521	102310	3096	8178	593
gcut13	3000×3000	≥13	≥ 8647565	-	-	-	-
cgcut1	15×10	8	244	44	11	14	1
cgcut2	40×70	12	2892	889	870	-	-
cgcut3	40×70	10	1860	582	227	356	102
okp1	100×100	11	27718	9467	234	3244	661
okp2	100×100	11	22502	72855	27585	23626	7310
okp3	100×100	11	24019	58161	8936	8233	816
okp4	100×100	10	32893	26088	4227	1458	15
okp5	100×100	8	27923	-	-	5733	643

Table 1: Number of nodes of the knapsack branch-and-bound tree and number of calls to the algorithm checking feasibility

The algorithm to check feasibility and the branch-and-bound procedure have been implemented in Java 6 and tested on a PC with a 3GHz Pentium IV processor. Runtimes reported by Fekete & Schepers in [19] were obtained from a similar PC, but their algorithm was developed in C++.

We used classical 2D-OKP benchmarks from [3, 11, 19] and the two dimensional guillotine cutting problems benchmarks from [2, 12]. All these benchmarks are available at

`http://www.ibr.cs.tu-bs.de/alg/packlib/instances.shtml`

Our binaries and the benchmarks files are also available at

`https://wiki.black.inria.fr/realopt/pmwiki.php/Project/SoftwareAlgoKP`

In tables 1 and 2, the columns JP (resp. FS, resp BB, resp. A0, A1, A2 and A3) correspond to our algorithm (resp. Fekete & Schepers' algorithm with data reported from [19], resp. Baldacci & Boschetti's algorithm with data reported from [1], resp. Caprara & Monaci's algorithms, as depicted in [8]). Table 1 gives the container sizes, the value of an optimal solution and the number of items of an optimal solution. The number of subset of items to be checked for feasibility (OKP nodes) and the number of effective calls (OPP nodes) to our algorithm to check feasibility (that is after the application of dual feasible functions) are reported.

Table 1 shows that the number of the OKP nodes of our algorithm is bigger than the number of OKP nodes of Fekete and Schepers' algorithm. This is due to the fact that our Branch-and-Bound is less sophisticated than Fekete and Schepers' to select a subset of items to be checked for feasibility. In particular, we do not make use of heuristics to try to guess a solution, in contrary to Fekete & Schepers'.

We report in Table 2 the running times of the algorithms. We also report the number of unsolved benchmarks (within the time limit of 1800 seconds) and the average time (computed on the set of instances `cgcut`, `gcut` and `okp`), with the convention that an unsolved benchmark counts for 1800 seconds. For `ngcut` and `hccut` instances, every algorithm is fast. `cgcut` and `gcut` instances are more difficult to solve, as the size of the container and/or the number of items of an optimal solutions are bigger. On these instances, our algorithm seems to be significantly faster, especially in the case of instances `cgcut2`, `gcut4`, `gcut8` and `gcut12`. This seems to indicate that we handle better symmetry issues due to items of same shape.

Results on the five `okp` instances are less conclusive: Fekete & Schepers' algorithm is faster (with the exception of `okp1`). Notice that on these instances, the number of calls to the algorithm to check feasibility is far much bigger in our case than in Fekete & Schepers'. Hence, as already noticed, Fekete & Scheper's branch-and-bound algorithm to handle the knapsack problem is more efficient. An explanation of this difference is that some symmetry issues are eliminated more efficiently by Fekete and Schepers' algorithm on these instances.

The instance `gcut13` is still open. We were able to provide a feasible solution of value 8647565 (involving 17 items), thus slightly improving Fekete & Schepers's lower bound of value 8622498 [19].

Benchmark	**JP**	**FS**	**BB**	**A0**	**A1**	**A2**	**A3**
ngcut1 . . . ngcut12	0	0	0	-	-	-	-
hccut2 . . . hccut4	0	0	0	-	-	-	-
cgcut1	0	0	0	0	1	1	1
cgcut2	**39**	>1800	>1800	>1800	>1800	533	531
cgcut3	0	0	95	23	23	4	4
gcut1	0	0	0	0	0	0	0
gcut2	0	0	0	0	0	25	0
gcut3	0	4	2	>1800	2	276	3
gcut4	**28**	195	46	>1800	346	>1800	376
gcut5	0	0	0	0	0	0	0
gcut6	0	0	1	0	0	9	0
gcut7	0	2	3	1	0	354	1
gcut8	**17**	255	186	1202	136	>1800	108
gcut9	0	0	0	0	0	0	0
gcut10	0	0	0	0	0	6	0
gcut11	**1**	8	3	16	14	>1800	16
gcut12	**3**	109	12	63	16	>1800	25
gcut13	>1800	>1800	>1800	>1800	>1800	>1800	>1800
okp1	**1**	10	779	24	25	72	35
okp2	477	20	288	>1800	>1800	1535	1559
okp3	7	5	0	21	1	465	10
okp4	23	2	14	40	2	0	4
okp5	>1800	11	190	40	>1800	513	488
# unsolved	2	2	2	5	4	5	1
Average time	200	200	248	496	369	609	239

Table 2: Running times (s)

4 Conclusion

In this paper, we gave an exact algorithm to solve the multi-dimensional orthogonal packing problem, which is based upon Fekete & Schepers' characterization of feasible packings by so-called packing classes. This algorithm is used to solve the two-dimensional orthogonal packing problem with two stages: the first stage is a basic branch-and-bound algorithm to select a subset of items and the second stage checks whether this subset of items admits a feasible packing. Our main contribution is for the second stage: to check feasibility, we used consecutive ones matrices as data structures to store feasible packings. In contrary to Fekete & Scheper's approach, by using consecutive ones matrices, we have a partial knowledge of the relative positions of the items and their coordinates. This enabled us to handle some symmetry issues in a close manner to the algorithms of Clautiaux, Moukrim and Carlier [13, 15].

Computer experiments on standard benchmarks show that this new algorithm is competitive. In a companion paper, we investigate the data structure of PQ-trees, as they were designed to store more efficiently the linear consecutive

ordering of the maximal cliques of interval graphs. Another matter of investigation is to derive from this algorithm an efficient mixed integer programming formulation for the multi-dimensional orthogonal knapsack problem. At last, this approach is suitable to solve other problems such as the orthogonal packing problem on a torus, by enumerating circulant consecutive-ones matrices.

References

1. Baldacci R, Boschetti MA (2007) A cutting plane approach for the two-dimensional orthogonal non-guillotine cutting stock problem. European Journal of Operational Research 183(3):1136–1149
2. Beasley JE (1985) Algorithms for unconstrained two-dimensional guillotine cutting. Journal of the Operational Research Society 36(4):297–306
3. Beasley JE (1985) An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research 33(1):49–64
4. Beasley JE, Mingozzi A (1996) A new formulation for the two-dimensional orthogonal cutting problem. Tech. rep., University of Bologna, Italy
5. Beldiceanu N, Carlsson M (2008) New filtering for the cumulative constraint in the context of non-overlapping rectangles. In: Lecture Notes in Computer Science, CPAIOR08, vol 5015, pp 21–25
6. Beldiceanu N, Carlsson M, Thiel S (2006) Sweep synchronization as a global propagation mechanism. COM 33(10):2835–2851
7. Below G, Scheithauer G (2008) Lower-dimensional bounds and a new model for higher-dimensional orthogonal packing
8. Caprara A, Monaci M (2004) On the two-dimensional knapsack problem. Operations Research Letters 32(1):5–14
9. Carlier J, Néron E (2000) A new lp-based lower bound for the cumulative scheduling problem. European Journal of Operational Research 127:363–382
10. Chen CS, Lee SM, Shen QS (1995) An analytical model for the container loading problem. European Journal of Operational Research 86(1):68–76
11. Christofides N, Hadjiconstantinou E (1995) An exact algorithm for orthogonal 2-d cutting problems using guillotine cuts. European Journal of Operational Research 83(1):21–38
12. Christofides N, Whitlock C (1977) An algorithm for two-dimensional cutting problems. Operations Research 25(1):30–44
13. Clautiaux F, Carlier J, Moukrim A (2007) A new exact method for the orthogonal packing problem. European Journal of Operational Research 183(3):1196–1211
14. Clautiaux F, Alves C, de Carvalho JV (2008) A survey of dual-feasible functions and superadditive functions. Annals of Operations Research
15. Clautiaux F, Jouglet A, Carlier J, Moukrim A (2008) A new constraint programming approach for the orthogonal packing problem. Computers and Operations Research 35(3):944–959
16. Dyckhoff H (1990) A typology of cutting and packing problems. European Journal of Operational Research 44(2):145–159
17. Fekete SP, Schepers J (2001) New classes of lower bounds for the bin packing problem. Mathematical Programming 91:11–31
18. Fekete SP, Schepers J (2004) A combinatorial characterization of higher-dimensional orthogonal packing. Mathematics of Operations Research 29(2):353–368
19. Fekete SP, Schepers J, van der Veen J (2007) An exact algorithm for higher-dimensional orthogonal packing. Operations Research 55(3):569–587
20. Ferreira E, Oliveira J (2008) Fekete and Schepers' Graph-based Algorithm for the Two-Dimensional Orthogonal Packing Problem Revisited. Gabler
21. Garey MR, Johnson DS (1979) Computers and intractability: A guide to the theory of np-completeness
22. Mercier L, v Hentenryck P (2008) Edge finding for cumulative scheduling. IJOC 20(1):143–153
23. Onodera H, Taniguchi Y, Tamaru K (1991) Branch-and-bound placement for building block layout. In: 28th ACM/IEEE design automation conference, pp 433–439
24. Padberg M (2000) Packing small boxes into a big box. Mathematical Methods Operations Research 52(1):1–21
25. Pisinger D, Sigurd M (2007) Using decomposition techniques and constraint programming for solving the two-dimensional bin packing problem. INFORMS Journal on Computing 19(1):36–51
26. Simonis H, O'Sullivan B (2008) Search strategies for rectangle packing
27. Wäscher G, HauBner H, Schuman H (2007) An improved typology of cutting and packing problems. European Journal of Operational Research 183(3):1109–1130