Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 36 (2010) 327–334
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

# Consecutive ones matrices for multi-dimensional orthogonal packing problems

Cédric Joncour [1]

*University of Bordeaux (LaBRI, INRIA), France*

Arnaud Pêcher [2]

*University of Toulouse (IRIT, INRIA), France*

**Abstract**

The multi-dimensional orthogonal packing problem (OPP) is a well studied optimization problem [3,9]. 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. Rotation of items is not allowed.

Fekete and Schepers introduced a tuple of interval graphs as data structures to store a feasible packing, and gave a very efficient algorithm. In this paper, we propose a new algorithm using consecutive one matrices as data structures, due to Fulkerson and Gross's characterization of interval graphs. Computational results are reported, which show its effectiveness.

*Keywords:* orthogonal packing problem, interval graph, consecutive ones matrices

The multi-dimensional orthogonal knapsack problem (OKP) is to compute the maximum value of a feasible set: if every item $i \in \mathcal{I}$ has a positive value

[1] Email:`cedric.joncour@math.u-bordeaux1.fr`

[2] Email:`arnaud.pecher@irit.fr`


doi:10.1016/j.endm.2010.05.042

$p_i$, the aim is to exhibit a subset of items $\mathcal{I}' \subset \mathcal{I}$ admitting a non-overlapping packing such that $\sum_{i \in \mathcal{I}'} p_i$ is maximal.

We consider the $D$-dimensional orthogonal packing problem and the $D$-dimensional orthogonal knapsack problem (with $D \geq 2$). Fekete and Schepers [9,10] 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 one 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 abstract is organized as follows. In the first section, we describe our model and algorithm. In the second section, we give computational results with respect to standard benchmarks.

## 1 Consecutive ones matrices for orthogonal packing

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

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

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

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):

$$x_i^d \in \{0\} \cup \left\{x_j^d + w_j^d : j \in \mathcal{I} \setminus \{i\}\right\} \quad \forall i \in \mathcal{I}, d \in \{1, \ldots, D\}$$

**Lemma 1.1** *For every feasible set of items $\mathcal{I}$, there is an associated feasible packing which verifies touching assumption.*

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 iff the corresponding intervals overlap.

Following Fekete and Schepers[9,10], a packing class is a collection of $D$ graphs $G_1, \dots, G_D$ with (shared) vertex set $\mathcal{I}$ and edge sets $E(G_d)$ such that:

**P1:** for every $d \in \{1, \dots, D\}$, $G_d$ is an interval graph;

**P2:** for every $d \in \{1, \dots, D\}$, for every stable set $S$ of $G_d$, $\sum_{s \in S} w_s^d \leq W^d$;

**P3:** $\displaystyle\bigcap_{d \in \{1,\dots,D\}} E(G_d) = \emptyset$.

Therefore to check whether a set of items is feasible, the algorithm of Fekete and Schepers enumerates all packing classes associated to $\mathcal{I}$. However, in some cases, there are distinct packings of a feasible set of items whose associated packing classes are different. Hence there are still symmetry issues in this model. A matrix $M \in \mathcal{M}_{n,m}(\mathbb{B})$ has the consecutive ones property if for every row, the set of 1s occur consecutively. We are now ready to state Fulkerson and Gross' characterization of interval graphs:

**Theorem 1.2** *[11] 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.*

Without loss of generality, let $\mathcal{I} = \{1, \dots, n\}$ be the set of items. From now on, $\forall d \in \{1, \dots, D\}$, let $m^d \leq n$, $M^d \in \mathcal{M}_{n,m^d}(\mathbb{B})$ be a matrix with consecutive ones property; $\forall k \in \{1, \dots, m^d\}$, let $Q_k^d = \{i \in \mathcal{I} : M_{ik}^d = 1\}$ be the set of items in column $k$ and let $C_k^d \in \mathbb{B}^n$ be the column $k$ of matrix $M^d$ representing the characteristic vector of $Q_k^d$; let $Q_{m^d+1}^d = \emptyset$ and let $\mathcal{Q}^d = \{Q_1^d, \dots, Q_{m^d}^d\}$.

We define the width $\lambda_k^d$ of $Q_k^d \in \mathcal{Q}^d, \forall d \in \{1, \dots, D\}$ by:

$$\lambda_k^d = \max_{\substack{i \in \mathcal{I} \\ i \in Q_k^d, i \notin Q_{k+1}^d}} \left\{ w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d \right\} \tag{3}$$

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

- for every dimension $d \in \{1, \dots, D\}$,

$$\text{every column of } M^d \text{ is maximal} \tag{4}$$

$\forall i \in \mathcal{I}$, $\forall k \in \{1, \dots, m^d\}$ such that $i \in Q_k^d$, we have

$$w_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d > 0 \tag{5}$$

$$\sum_{k \in \{1,\dots,m^d\}} \lambda_k^d \leq W^d \tag{6}$$

- for every pair $i, j \in \mathcal{I} (i \neq j)$, there is a dimension $d \in \{1, \ldots, D\}$ such that,

$$\forall k \in \{1, \ldots, m^d\},\ i \notin Q_k^d \text{ or } j \notin Q_k^d \tag{7}$$

**Lemma 1.3** *For every $i \in \mathcal{I}$ and $d \in \{1, \ldots, D\}$, $w_i^d \leq \sum_{1 \leq k \leq m^d / i \in Q_k^d} \lambda_k^d$*

**Lemma 1.4** *A set of items $\mathcal{I}$ is feasible with touching assumption if and only if there is a strip decomposition associated to $\mathcal{I}$.*

Due to Lemma 1.4, to check feasibility, we only have to design an algorithm which returns whether there is a strip decomposition associated to the set of items $\mathcal{I}$. We proceed dimension by dimension, by enumerating all consecutive ones matrices which satisfy the strip decomposition constraints (4), (5), (6) and (7). This is a main point of our approach: almost all the work is done in first dimension.

To ensure that constraint (7), is satisfied, we proceed like this: for every dimension $d \in \{1, \ldots, D-1\}$, let $S_d$ denote the set of all matrices output by the recursion w.r.t. dimension $d$. Then for the last dimension $d = D$, for every $(D-1)$-tuple of matrices $(M^1, \ldots, M^{D-1}) \in S_1 \times \ldots \times S_{D-1}$, we look for a consecutive ones matrix such that constraint (7) 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.

### *1.1 Early detection of unfeasibility*

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

Assume that the first $k$ columns of the matrix in the recursion are set. Every item which has only 0 entries in these columns has to be packed in the remaining columns. Therefore the following constraint, which says that the available width must exceed the biggest width of items that are not packed yet, is a valid constraint:

**Lemma 1.5** *For every $d \in \{1, \ldots, D\}$ and $k \in \{1, \ldots, m^d\}$,*

$$\sum_{l \in \{1, \ldots, k\}} \lambda_l^d + \max_{i \notin \{Q_1^d, \ldots, Q_k^d\}} w_i^d \leq W^d \tag{8}$$

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 1.6** *For every $d \in \{1, \ldots, D\}$ and $k \in \{1, \ldots, m^d\}$,*

$$\sum_{i \in Q_k^d} \Bigg( \prod_{\substack{d' \in \{1,\ldots,D\} \\ d' \neq d}} w_i^{d'} \Bigg) \leq \prod_{\substack{d' \in \{1,\ldots,D\} \\ d' \neq d}} W^{d'} \tag{9}$$

Assume that the first $k$ columns in the recursion 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:

**Lemma 1.7** *For every $d \in \{1, \ldots, D\}$ and $k \in \{1, \ldots, m^d\}$,*

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

$$+ \sum_{i \notin \{Q_1^d,\ldots,Q_k^d\}} \Bigg( \prod_{d' \in \{1,\ldots,D\}} w_i^{d'} \Bigg) \leq \prod_{d' \in \{1,\ldots,D\}} W^{d'} \tag{10}$$

### *1.2 Breaking some symmetry issues*

The first solution which is found by our recursive algorithm is the biggest matrix satisfying strip decomposition constraints without 0 rows, with respect to the lexicographic order. Hence, if we are able during the recursion to detect that if the packing is feasible, then a solution with biggest lexicographic order should have already been found, then we can safely stop it and return that the packing is infeasible. We defined several data structures (which are described in the full paper) in order to early detect such configurations: these greatly improved the efficiency of the algorithm.

## 2 Computational results

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 basic branch-and-bound algorithm, calling our algorithm to check 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 [10] were obtained from a similar PC, but their algorithm was developed in C++.

We used classical 2D-OKP benchmarks from [3,5,10] and the two dimensional guillotine cutting problems benchmarks from [2,6].

In Table 1, the column JP (resp. FS, resp BB, resp. A0, A1, A2 and A3) corresponds to our algorithm (resp. Fekete & Schepers' algorithm with data reported from [10], resp. Baldacci & Boschetti's algorithm with data reported from [1], resp. Caprara & Monaci's algorithms, as depicted in [4]).

We report in Table 1 the running times of the algorithms. On `cgcut` and `gcut` 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 better handle 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`). 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, Fekete & Scheper's branch-and-bound algorithm to handle the knapsack problem is more efficient.

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 [10].

## 3 Conclusion

In this paper, we gave an exact algorithm to solve the multi-dimensional orthogonal knapsack problem, which is based upon Fekete & Schepers' characterization of feasible packings by so-called packing classes. This algorithm has 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 [7,8].

Computer experiments on standard benchmarks show that this new algorithm is competitive. For further work, data structure such as PQ-trees

| **Benchmark** | **JP** | **FS** | **BB** | **A0** | **A1** | **A2** | **A3** |
|---|---|---|---|---|---|---|---|
| ngcut1 ... ngcut11 | 0 | 0 | 0 | | | | |
| hccut2 ... hccut5 | 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 |

Table 1
Running times ($s$)

should be investigated, 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.

## References

[1] Baldacci, R. and M. A. Boschetti, *A cutting plane approach for the two-dimensional orthogonal non-guillotine cutting stock problem*, European Journal of Operational Research **183** (2007), pp. 1136–1149.

[2] Beasley, J. E., *Algorithms for unconstrained two-dimensional guillotine cutting*, Journal of the Operational Research Society **36** (1985), pp. 297–306.

[3] Beasley, J. E., *An exact two-dimensional non-guillotine cutting tree search procedure*, Operations Research **33** (1985), pp. 49–64.

[4] Caprara, A. and M. Monaci, *On the two-dimensional knapsack problem*, Operations Research Letters **32** (2004), pp. 5–14.

[5] Christofides, N. and E. Hadjiconstantinou, *An exact algorithm for orthogonal 2-d cutting problems using guillotine cuts*, European Journal of Operational Research **83** (1995), pp. 21–38.

[6] Christofides, N. and C. Whitlock, *An algorithm for two-dimensional cutting problems*, Operations Research **25** (1977), pp. 30–44.

[7] Clautiaux, F., J. Carlier and A. Moukrim, *A new exact method for the orthogonal packing problem*, European Journal of Operational Research **183** (2007), pp. 1196–1211.

[8] Clautiaux, F., A. Jouglet, J. Carlier and A. Moukrim, *A new constraint programming approach for the orthogonal packing problem*, Computers and Operations Research **35** (2008), pp. 944–959.

[9] Fekete, S. P. and J. Schepers, *A combinatorial characterization of higher-dimensional orthogonal packing*, Mathematics of Operations Research **29** (2004), pp. 353–368.

[10] Fekete, S. P., J. Schepers and J. van der Veen, *An exact algorithm for higher-dimensional orthogonal packing*, Operations Research **55** (2007), pp. 569–587.

[11] Fulkerson, D. R. and O. A. Gross, *Incidence matrices and interval graphs*, Pacific Journal of Mathematics **15** (1965), pp. 835–855.