Istituto di Analisi dei Sistemi ed Informatica
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

T. Bacci, S. Nicoloso

ON THE BENCHMARK INSTANCES FOR THE BIN PACKING WITH CONFLICTS

R. 4, 2018

Tiziano Bacci – IASI - CNR — Via dei Taurini 19, 00185 Roma, Italia (tiziano.bacci@iasi.cnr.it).

Sara Nicoloso – IASI - CNR — Via dei Taurini 19, 00185 Roma, Italia (sara.nicoloso@iasi.cnr.it).

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

via dei Taurini 19, 00185 ROMA, Italy

tel. ++39-06-49937101/02
fax ++39-06-49937106
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

Many authors, mainly in the context of the Bin Packing Problem with Conflicts, used the random graph generator proposed in "Heuristics and lower bounds for the bin packing problem with conflicts" [M. Gendreau, G. Laporte, and F. Semet, *Computers & Operations Research*, 31:347–358, 2004]. In this paper we prove that the graphs generated in this way are not arbitrary but threshold ones. Computational results show that instances of the Bin Packing Problem with Conflicts on threshold graphs are easier to solve w.r.t. instances on arbitrary graphs.

Key words: Bin Packing with Conflicts, threshold graphs, random graph generator

1. Introduction

In this paper we show that a popular random graph generator Gendreau et al. (2004), widely used in the context of Bin Packing Problem with Conflicts, generates very special graphs namely threshold graphs and not arbitrary ones as claimed by Gendreau et al. (2004), nor arbitrary interval ones as claimed by Sadykov and Vanderbeck (2013).

In Section 2 we recall the definition of threshold graphs and discuss some of their peculiar properties, in Section 3 we present the generator defined in Gendreau et al. (2004) showing that it produces threshold graphs, in Section 4 we analyse the effects of using this generator on instances of Bin Packing Problem with Conflicts. Concluding remarks in Section 5.

2. Threshold graphs

A graph $G = (V, E)$ is a threshold graph if there exist a real number d (the threshold) and a weight p_x for every vertex $x \in V$ such that (i, j) is an edge iff $(p_i + p_j)/2 \leq d$ (see Golumbic (1980)). W.l.o.g. from now on we assume that $p_x \in [0, 1]\ \forall x$ (as a consequence it makes sense to choose $d \in [0, 1]$).

According to this definition it follows that a vertex i is connected to all the vertices j such that $p_j \leq 2d - p_i$. Thus, $N(h) \supseteq N(k)$ and $\deg(h) \geq \deg(k)$ if and only if $p_h \leq p_k$, where $N(x)$ denotes the set of vertices adjacent to x and $\deg(x) = |N(x)|$.

A threshold graph has many peculiar properties as it is at the same time an interval graph, a co-interval graph, a cograph, a split graph, and a permutation. In addition, its complement, where (i, j) is an edge iff $(p_i + p_j)/2 > d$, is a threshold graph too.

W.l.o.g. from now on we assume that the vertices of a threshold graph G are numbered in such a way that $i < j$ if and only if $\deg(i) \geq \deg(j)$. Then the $n \times n$ adjacency matrix $M = [m_{i,j}]$ of G always appears as in Figure 1, where an entry 0 is coloured in white and an entry 1 is highlighted in grey, and $m_{i,i} = 0$ for $i = 1, \ldots, n$.

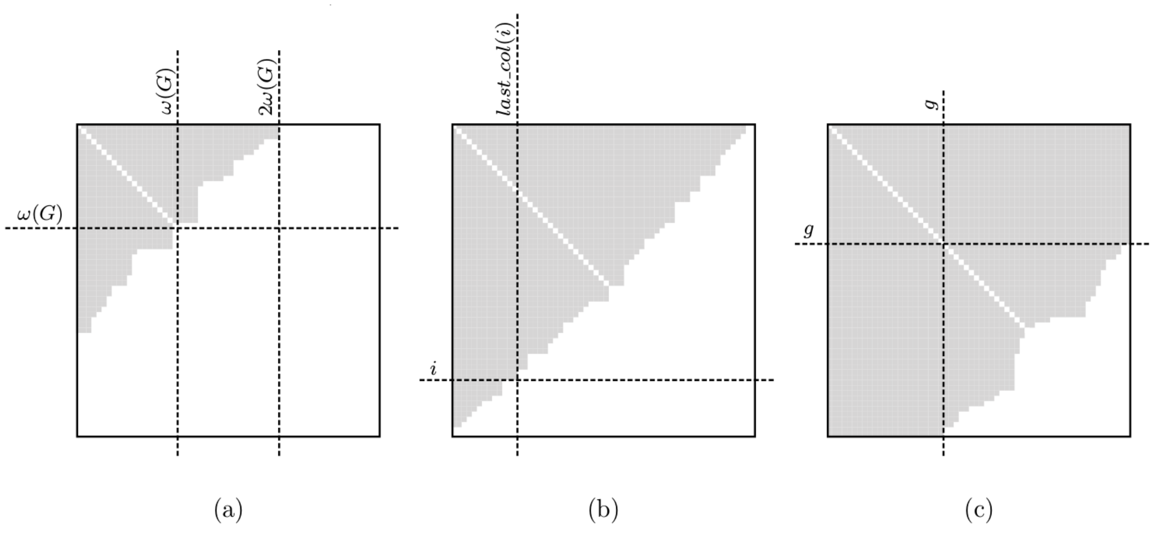


Figure 1: Examples of adjacency matrices of threshold graphs with $n = 60$ nodes and threshold a) $d = 0.2$, b) $d = 0.5$, and c) $d = 0.8$.

By what above, we observe what follows.

4.

1. For each row i, let $last\_col(i) = \max\{j : m_{i,j} = 1, j = 1, \dots, n\}$ if $m_{i,1} = 1$, and $last\_col(i) = 0$ if $m_{i,1} = 0$ (see Figure 1b); hence $last\_col(i) \geq last\_col(i+1)$.

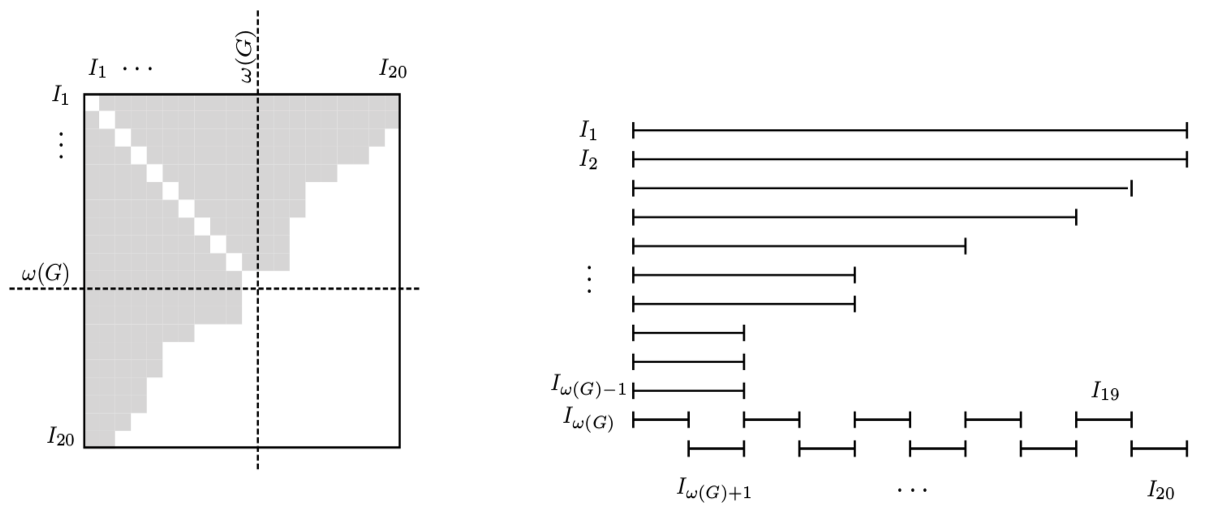
2. Let $t = \min\{j : m_{j,j+1} = 0, j = 1, \dots, n\}$. Then the set of vertices $\{1, \dots, t\}$ induces a maximum clique of size $\omega(G) = t$ (see Figure 1a). In fact, by definition, $m_{t-1,t} = 1$, thus $last\_col(t-1) \geq t$ and, by Point 1, $m_{i,j} = 1$ for $i = 1, \dots, t$ and $j = 1, \dots, t$, $i \neq j$.

3. The set of vertices $\{t, \dots, n\}$ induces a maximum independent set of size $n - t + 1$. In fact, by definition, $m_{t,t+1} = 0$ and $m_{t,t-1} = 1$ (as $m_{t-1,t} = 1$) thus $last\_col(t) = t - 1$ and $m_{i,j} = 0$ for $i = t, \dots, n$ and $j = t, \dots, n$ (see Point 1.).

4. Let $g = \max\{h : m_{h,n} = 1, h = 1, \dots, n\}$ if $last\_col(1) = n$, and $g = 0$ otherwise (see Figure 1c). Clearly, $g = last\_col(n)$. If $g \geq 1$, vertex i, for $i = 1, \dots, g$, is connected to any other vertex.

5. Recalling that a threshold graph G is a particular interval graph, it is always possible to derive the following family of intervals whose intersection graph is G: to each vertex $j = t, \dots, n$, associate the interval $I_j = (l_j, r_j) = (j - t, j - t + 1)$; to each vertex $j = 1, \dots, t-1$, associate the interval $I_j = (l_j, r_j) = (0, r_{last\_col(j)}) = (0, last\_col(j) - t + 1)$ (we remark that $r_j \geq 1$ as $last\_col(j) \geq t\}$). See an example in Figure 2.

6. The edge density $\delta = 2|E|/(n(n-1))$ of G is not equal to the threshold d, generally speaking.



(a) (b)

Figure 2: a) The adjacency matrix of threshold graph with $n = 20$ nodes and b) the corresponding interval model.

For $n \to \infty$ and $p_1, \dots, p_n$ uniformly distributed in $[0, 1]$, one has:

7. $\omega(G) = t = nd$.

8. The edge density $\delta = 2|E|/(n(n-1))$ of G depends on d. Precisely

$$\delta = f(d) = \begin{cases} \frac{2(nd)^2 - nd}{n(n-1)} & \text{for } d \leq 0.5 \\ \\ \frac{n(n-1) - 2n^2(1-d)^2 - n(1-d)}{n(n-1)} & \text{for } d \geq 0.5 \end{cases}$$

   In fact, for $d \leq 0.5$ the $2|E|$ 1's are in the area $A \cup B \cup C$ (see Figure 3a). In a similar way one can compute the number of 1's in the matrix when $d \geq 0.5$.

9. $g = 0$ when $d \leq 0.5$, and $g = n(2d-1)$ when $d \geq 0.5$ (see Figure 3b).

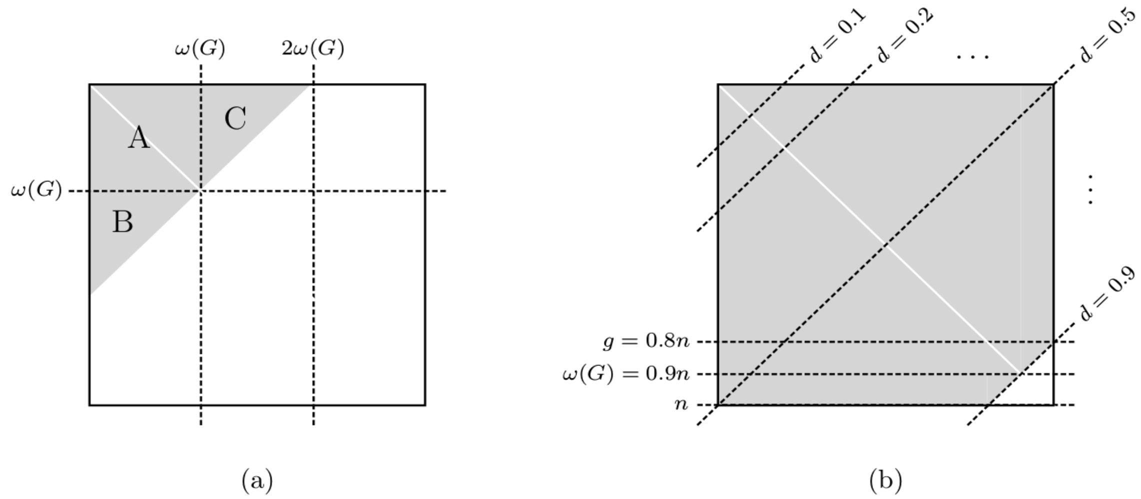


Figure 3: The expected adjacency matrix when $n \to \infty$ and with threshold a) $d \leq 0.5$ and b) $d \geq 0.5$

3. A random threshold graph generator

Gendreau et al. (2004) describe the following generator: "*A value p_i was first assigned to each vertex $i \in V$ according to a continuous uniform distribution on $[0, 1]$. Each edge (i, j) of G was created whenever $(p_i + p_j)/2 \leq d$, where d is the expected density of G.*".

This generator clearly produces threshold graphs, and the expected edge density of G is not d as claimed but it is the one discussed in Points 6 and 8 of Section 2. To get a threshold graph with expected edge density δ one has to set

$$d = \begin{cases} \frac{1+\sqrt{1+8n(n-1)\delta}}{4n} & \text{for } \delta \leq 0.5 \\ \\ 1 + \frac{1-\sqrt{1+8n(n-1)(1-\delta)}}{4n} & \text{for } \delta \geq 0.5 \end{cases}$$

Already for $n \geq 100$ these values can be approximated to $d = \sqrt{\delta/2}$ and $d = 1 - \sqrt{(1-\delta)/2}$, respectively.

The generator by Gendreau et al. (2004) has been improperly used to generate arbitrary graphs Basnet and Wilson (2005); Brandão and Pedroso (2016); Capua et al. (2015); Clautiaux et al. (2011); Cornaz et al. (2017); Elhedhli et al. (2011); Gschwind and Irnich (2016); Joncour (2010); Joncour et al. (2010); Jouida et al. (2015); Khanafer (2010); Khanafer et al. (2012a, 2010, 2012b); Maiza and Guéret (2009); Maiza and Radjef (2011); Muritiba (2010); Muritiba et al. (2010); Sadykov and Vanderbeck (2013); Yuan et al. (2014). In particular, Muritiba et al. (2010) made publicly available "benchmark" instances generated in this way (see http://or.dei.unibo.it/library/bin-packing-problem-conflicts) and used by many authors Brandão and Pedroso (2016); Capua et al. (2015); Clautiaux et al. (2011); Cornaz et al. (2017); Elhedhli et al. (2011); Gschwind and Irnich (2016); Joncour (2010); Joncour et al. (2010); Khanafer (2010); Khanafer et al. (2012a, 2010); Muritiba (2010); Sadykov and Vanderbeck (2013); Yuan et al. (2014).

Most of the authors using the generator by Gendreau et al. (2004) claim that they group the graphs of their test bed by edge densities, but actually they group the graphs by threshold values. Our analysis on the instances by Muritiba et al. (2010) shows that the relation between the threshold d and the corresponding edge density δ is the following.

d	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
δ	0	0.02	0.08	0.18	0.32	0.5	0.68	0.82	0.92	0.98

We remark that the values of δ coincide with those which can be computed by the formula of Point 8 in Section 2.

4. Bin Packing Problem with Conflicts on threshold graphs

The Bin Packing Problem with Conflicts ($BPPC$), first introduced in a scheduling context by Jansen and Öhring (1997), is defined as follows: given a graph $G = (V, E)$, a nonnegative integer weight w_i for each vertex $i \in V$, and a nonnegative integer B, find a partition of V into k subsets $V_1, \ldots, V_k$, such that the sum of the weights of the vertices assigned to same subset is less than or equal to B, two vertices connected by an edge do not belong to the same subset, and k is minimum.

The minimum value of k will be denoted k_{BPPC}. The graph $G = (V, E)$ is called *conflict graph* and two vertices connected by an edge are said to be *in conflict*.

$BPPC$ generalizes two well known combinatorial optimization problems, the Bin Packing Problem and the Vertex Coloring Problem. In fact, $BPPC$ reduces to Bin Packing when the edge set E of the graph G is empty, and it reduces to Vertex Coloring when $B \geq \sum_{i \in V} w_i$ or when G is complete. Observe that Vertex Coloring is solvable in linear time on threshold graphs, nevertheless $BPPC$ with a threshold conflict graph is NP-hard because Bin Packing is (Garey and Johnson (1979)).

Since threshold graphs are a subclass of interval graphs, which are in their turn a subclass of arbitrary graphs, we expect that $BPPC$ on threshold graphs is the easiest to solve. To prove our claim we conducted some computational experiments.

By $X(n, \delta)$ with $X \in \{T, I, A\}$ we denote a set of ten instances with n vertices, bound $B = 150$, and conflict graph which is a threshold graph if $X = T$, an interval graph if $X = I$, an arbitrary graph if $X = A$, with expected edge density $\delta \in \{0.02, 0.08, 0.18, 0.32, 0.5, 0.68, 0.82, 0.92, 0.98\}$

(the same densities of the instances used in Muritiba et al. (2010)). In particular, we choose $n \in \{250, 1000\}$.

The $T(250, \delta)$ and $T(1000, \delta)$ instances are exactly those in the classes 2 and 4 by Muritiba et al. (2010), respectively. Precisely, given n, the weight of the i-th vertex of the k-th instance of $T(n, \delta)$ is the same for all δ. Totally we consider 180 out of 800 of the instances by Muritiba et al.

As for the $I(n, \delta)$, the weight of the i-th vertex of the k-th instance is exactly the weight of the i-th vertex of the k-th instance of $T(n, \delta)$, and the arbitrary interval conflict graphs have

		n = 250						n = 1000					
		T		I		A		T		I		A	
		Opt	Time	Opt	Time	Opt	Time	Opt	Time	Opt	Time	Opt	Time
δ	0.02	10	1.28	1	138.28	1	206.62	10	76.37	0	-	0	-
	0.08	10	2.75	0	-	0	-	10	292.26	0	-	0	-
	0.18	10	3.37	1	522.39	0	-	10	359.16	0	-	0	-
	0.32	10	3.81	10	201.56	4	340.2	3	444.77	0	-	0	-
	0.5	10	1.00	10	15.31	10	75.16	10	390.94	0	-	0	-
	0.68	10	0.53	10	3.24	10	12.43	10	294.12	0	-	0	-
	0.82	10	0.29	10	2.02	10	5.15	10	222.57	5	543.92	0	-
	0.92	10	0.11	10	1.39	10	2.89	10	197.60	10	453.98	0	-
	0.98	10	0.04	10	1.03	10	1.92	10	199.36	10	366.07	3	561.02

Table 1: Computational results on instances with threshold (T), interval (I)[1], and arbitrary (A) conflict graphs.

been generated according to the interval graph generator by Bacci and Nicoloso (2017)[1].

As for the $A(n, \delta)$, the weight of the i-th vertex of the k-th instance is exactly the weight of the i-th vertex of the k-th instance of $T(n, \delta)$, and the arbitrary conflict graphs have been generated as in Sadykov and Vanderbeck (2013): "*We began with the empty graph. We iteratively selected an item pair (i, j) at random (with uniform distribution); then edge (i, j) was added to the graph if it was not already defined. The procedure was interrupted as soon as the desired graph density was reached.*".

We solve to optimality the $T(n, \delta)$, $I(n, \delta)$, and $A(n, \delta)$ instances for all n and δ by means of the Vector Packing Solver 3.1.2 (VPS for short) by Brandão and Pedroso (2016), available at http://vpsolver.dcc.fc.up.pt/. This method is based on an arc-flow formulation with side constraints and builds very strong integer programming models that can be given in input to any state-of-the-art mixed integer programming solver (we used Cplex 12.6 on an Intel Core i7-3632QM 2.20GHz $\times$ 8 with 16 GB RAM under a Linux operating system). Actually, the arc-flow formulation is derived from a suitable graph which is preliminarily generated and whose size increases rapidly with B. We remark that the algorithm is applied to many classical combinatorial problems: in particular, it is one of the best behaving exact approaches for the instances by Muritiba et al. (2010), which are all solved to optimality within 50 minutes and with an average runtime of two minutes. In our analysis we set a time limit of 600 seconds for each instance.

The computational results are summarized in Table 1, where rows are indexed by δ, and columns by the type of the conflict graph. In the "Opt" columns we report the number of instances, out of ten, solved to optimality within the time limit, and in the "Time" columns the time in seconds required to solve one instance, averaged over the solved instances, only.

The results in the table show that threshold instances T are easier w.r.t. instances with interval conflict graphs, and the latter are easier than instances with arbitrary conflict graphs, confirming our claim.

We remark that, as far as we know, no tests on instances of $BPPC$ with arbitrary interval conflict graphs were performed in the literature. Sadykov and Vanderbeck (2013) observe that the conflict graphs of the benchmark instances by Muritiba et al. (2010) are interval graphs and not arbitrary graphs (actually they are not arbitrary interval ones). Nevertheless, to our

[1] The generator in Bacci and Nicoloso (2017) is not able to produce interval graphs with $n = 1000$ and edge density $\delta = 0.98$; in the corresponding cell of Table 1 of the present paper the average edge density of the ten instances is 0.96.

knowledge, Sadykov and Vanderbeck (2013) are the only ones who test their algorithm on instances with arbitrary conflict graphs.

5. Concluding remarks

In this paper we show that graphs of the $BPPC$ instances considered in Basnet and Wilson (2005); Brandão and Pedroso (2016); Capua et al. (2015); Clautiaux et al. (2011); Cornaz et al. (2017); Elhedhli et al. (2011); Gschwind and Irnich (2016); Joncour (2010); Joncour et al. (2010); Jouida et al. (2015); Khanafer (2010); Khanafer et al. (2012a, 2010, 2012b); Maiza and Guéret (2009); Maiza and Radjef (2011); Muritiba (2010); Muritiba et al. (2010); Sadykov and Vanderbeck (2013); Yuan et al. (2014) and generated according to Gendreau et al. (2004) are threshold graphs (and not arbitrary ones), and their edge density is not the declared one. Computational evidence suggests that $BPPC$ instances with threshold conflict graphs are easier to solve than instances with interval or arbitrary conflict graphs.

A consequence of using the generator by Gendreau et al. (2004) in the context of $BPPC$ is that, according to Point 4 in Section 2, when $d \geq 0.5$, in any optimal solution $V_i = \{i\}$ for $i = 1, \dots, g$, and V_i for $i \geq g + 1$ can be determined by solving a smaller instance $\mathcal{Q}$ defined on the last $n - g$ vertices (observe that the problem becomes simpler and simpler as d increases). The conflict graph of $\mathcal{Q}$ is a threshold graph with expected edge density 0.5, and contains a maximum clique of expected size $(n - g)/2$. So the expected value of a lower bound for k_{BPPC} on the initial instance is $g + (n - g)/2$. For example, when $n = 120$ and $d = 0.9$, g is expected to be $0.8n = 96$, $\mathcal{Q}$ has 24 vertices, on average, and $k_{BPPC} \geq 96 + 12 = 108$ (this value appears in Table 2, column LBO, Size 120, $d = 90$ of Elhedhli et al. (2011)).

Finally we remark that Gendreau et al. (2004) claim to use "the procedure described in" Soriano and Gendreau (1996), but this is not true. In fact, the procedure by Soriano and Gendreau (1996) generates "*edge (i, j) with probability*" $(p_i + p_j)/2$ generalizing the uniform random graph generator and outputing arbitrary graphs.

References

T. Bacci and S. Nicoloso. A heuristic algorithm for the Bin Packing Problem with Conflicts on Interval Graphs, 2017. arXiv:1707.00496 [math.CO].

C. Basnet and J. Wilson. Heuristics for determining the number of warehouses for storing non-compatible products. *International Transactions in Operations Research*, 12:527–538, 2005.

F. Brandão and J. P. Pedroso. Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, 69:56–67, 2016.

R. Capua, Y. Frota, T. Vidal, and L. S. Ochi. Um algoritmo heurìstico para o problema de bin packing com conflitos. Manuscript, 2015.

F. Clautiaux, A. Khanafer, S. Hanafi, and E. Talbi. Le problème bi-objectif de bin-packing avec conflits. Manuscript, 2011.

10.

D. Cornaz, F. Furini, and E. Malaguti. Solving vertex coloring problems as maximum weight stable set problems. *Discrete Applied Mathematics*, 217:151 – 162, 2017.

S. Elhedhli, L. Li, M. Gzara, and J. Naoum-Sawaya. A Branch-and-Price algorithm for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 23:404–415, 2011.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Co, New York, 1979.

M. Gendreau, G. Laporte, and F. Semet. Heuristics and lower bounds for the bin packing problem with conflicts. *Computers & Operations Research*, 31:347–358, 2004.

M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, 1980.

T. Gschwind and S. Irnich. Dual inequalities for stabilized column generation revisited. *INFORMS Journal on Computing*, 28(1):175–194, 2016.

K. Jansen and S. Öhring. Approximation algorithms for time constrained scheduling. *Information and Computation*, 132(2):85 – 108, 1997.

C. Joncour. *Problèmes de placement 2D et application à l'ordonnancement: modélisation par la théorie des graphes et approches de programmation mathématique.* PhD thesis, Université de Bordeaux I, July 2010. Number 4173.

C. Joncour, S. Michel, R. Sadykov, D. Sverdlov, and F. Vanderbeck. Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, 36:695–702, 2010.

S. B. Jouida, A. Ouni, and S. Krichen. A multi-start tabu search based algorithm for solving the warehousing problem with conflict. In H. A. Le Thi, T. Pham Dinh, and N. T. Nguyen, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences: Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences - MCO 2015 - Part II*, pages 117–128. Springer International Publishing, 2015.

A. Khanafer. *Algorithmes pour des problèmes de bin packing mono- et multi-objectif.* PhD thesis, Université Lille1, 2010. Number 40363.

A. Khanafer, F. Clautiaux, S. Hanafi, and E. Talbi. The min-conflict packing problem. *Computers & Operations Research*, 39:2122–2132, 2012a.

A. Khanafer, F. Clautiaux, and E. Talbi. New lower bounds for bin packing problems with conflicts. *European Journal of Operational Research*, 206(2):281–288, 2010.

A. Khanafer, F. Clautiaux, and E. Talbi. Tree-decomposition based heuristics for the two-dimensional bin packing problem with conflicts. *Computers & Operations Research*, 39:54–63, 2012b.

M. Maiza and C. Guéret. A new lower bound for bin packing problem with general conflicts graph. In *1st Doctoriales STIC'09*, Université de M'sila, Algérie, 2009.

M. Maiza and M. S. Radjef. Heuristics for solving the bin-packing problem with conflicts. *Applied Mathematical Sciences*, 5:1739 – 1752, 2011.

F. A. E. Muritiba. *Algorithms and Models For Combinatorial Optimization Problems*. PhD thesis, Alma Mater Studiorum Università di Bologna, 2010.

F. A. E. Muritiba, M. Iori, E. Malaguti, and P. Toth. Algorithms for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 22(3):401–415, 2010.

R. Sadykov and F. Vanderbeck. Bin packing with conflicts: a generic Branch-and-Price algorithm. *INFORMS Journal on Computing*, 25(2):244–255, 2013.

P. Soriano and M. Gendreau. Tabu search algorithms for the maximum clique problem. *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*, 26(2):221–236, 1996.

Y. Yuan, Y. Li, and Y. Wang. An improved aco algorithm for the bin packing problem with conflicts based on graph coloring model. *21th International Conference on Management Science & Engineering, Helsinki, Finland*, 2014.