On the Benchmark Instances for the Bin Packing Problem with Conflicts

Tiziano Bacci and Sara Nicoloso

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 show 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.

Keywords Bin packing with conflicts · Threshold graphs · Random graph generator

1 Introduction

The Bin Packing Problem with Conflicts ($BPPC$), first introduced in a scheduling context in [14], 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 $V_1, \dots, V_k$, 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

T. Bacci (✉) · S. Nicoloso
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", C.N.R., Rome, Italy
e-mail: tiziano.bacci@iasi.cnr.it; sara.nicoloso@iasi.cnr.it

C. Gentile et al. (eds.), *Graphs and Combinatorial Optimization: from Theory to Applications*, AIRO Springer Series 5,
https://doi.org/10.1007/978-3-030-63072-0_14

is solvable in linear time on threshold graphs, nevertheless $BPPC$ with a threshold conflict graph is NP-hard because Bin Packing is [10].

In this paper we show that a popular random graph generator [11], widely used in the context of the Bin Packing Problem with Conflicts, generates threshold graphs. Threshold graphs are very special interval graphs, contradicting the authors [11] who claim that "No assumptions are made on the adjacency structure of the graph", and strengthening [28] where the authors recognise the graphs as being arbitrary interval graphs [28].

In Sect. 2 we recall the definition of threshold graphs and discuss some of their peculiar properties, in Sect. 3 we present the generator defined in [11] showing that it produces threshold graphs, in Sect. 4 we analyse the effects of using this generator on instances of Bin Packing Problem with Conflicts. Concluding remarks are discussed in Sect. 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 $(p_i + p_j)/2 \le d$ (see [12]). 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 $p_j \le 2d - p_i$. Let $N(x)$ denote the set of vertices adjacent to x and let $\deg(x) = |N(x)|$. Then $N(h) \supseteq N(k)$ and $\deg(h) \ge \deg(k)$ if and only if $p_h \le p_k$.

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 graph. 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 $\deg(i) \ge \deg(j)$. Then the $n \times n$ symmetric adjacency matrix $M = [m_{i,j}]$ of G always appears as in Fig. 1, where an entry 0 is coloured in white and an entry 1 is highlighted in grey, and $m_{i,i} = 0$ for $i = 1, \dots, n$.

By what above, we observe what follows.

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 Fig. 1b); hence $last\_col(i) \ge last\_col(i + 1)$.
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 Fig. 1a). In fact, by definition, $m_{t-1,t} = 1$, thus $last\_col(t - 1) \ge t$ and, by Point 1, $m_{i,j} = 1$ for $i = 1, \dots, t$ and $j = 1, \dots, t$, $i \ne j$.

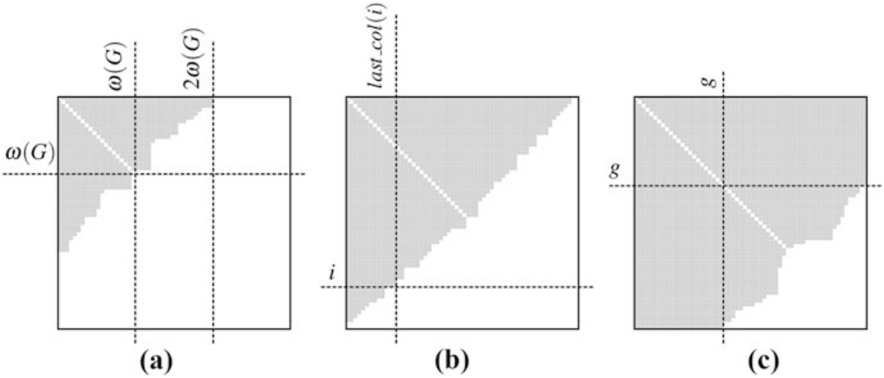

Fig. 1 Examples of adjacency matrices of threshold graphs with $n = 60$ nodes and threshold (**a**) $d = 0.2$, (**b**) $d = 0.5$, and (**c**) $d = 0.7$

3. The set of vertices $\{t, \ldots, n\}$ 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 $i = t, \ldots, n$ and $j = t, \ldots, n$ (see Point 1.).
4. Let $g = last\_col(n)$ (see Fig. 1c). If $g \geq 1$, vertex i, for $i = 1, \ldots, g$, is connected to any other vertex.
5. Recalling that a threshold graph G is a particular interval graph, it is always possible to derive a family of (open) intervals whose intersection graph is G, namely: to each vertex $j = t, \ldots, n$, associate the interval $I_j = (l_j, r_j) = (j - t, j - t + 1)$; to each vertex $j = 1, \ldots, t - 1$, 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 Fig. 2.
6. The edge density $\delta = 2|E|/(n(n-1))$ of G is not equal to the threshold d, generally speaking.

For $n \to \infty$ and $p_1, \ldots, p_n$ 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 Fig. 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 Fig. 3b).

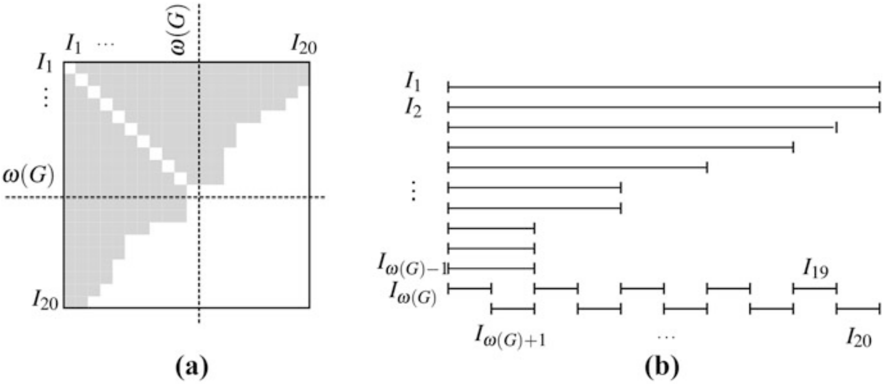

Fig. 2 (**a**) The adjacency matrix of threshold graph with $n = 20$ nodes and (**b**) the corresponding opened interval model

According to Points 2 and 3 above, when $d \geq 0.5$, in any optimal solution of $BPPC$ one has $V_i = \{i\}$ for $i = 1, \ldots, g$. The remaining sets V_i for $i \geq g + 1$ can be determined by solving a smaller instance Q defined on the last $n - g$ vertices (observe that the problem becomes simpler and simpler as d increases). The conflict graph of Q is a threshold graph with expected edge density 0.5, thus it contains a maximum clique of expected size $(n - g)/2$. According to Point 7, when $n = 120$ and $d = 0.9$, a lower bound for k_{BPPC} is $nd = 108$ (indeed this value appears in Table 2, column LBO, Size 120, $d = 90$ in [9]).

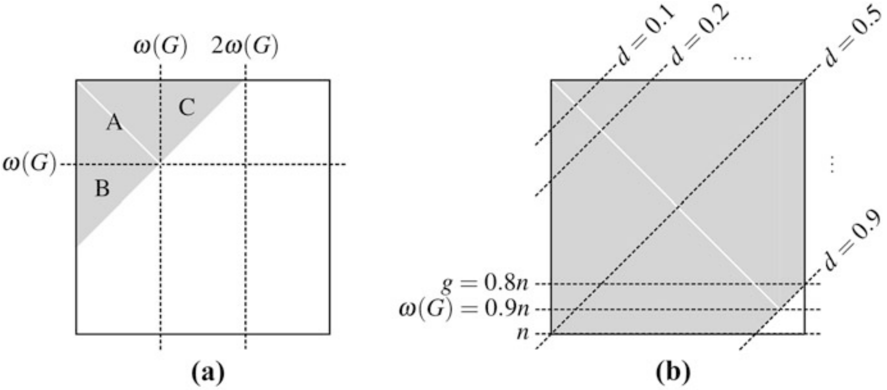

Fig. 3 The expected adjacency matrix when $n \to \infty$ and with threshold (**a**) $d \leq 0.5$ and (**b**) $d \geq 0.5$

3 On the Random Graph Generator Proposed in [11]

In [11] the following random graph generator is described: "*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 a threshold graph whose expected edge density is not d as claimed but it is the one discussed in Points 6 and 8 of Sect. 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 in [11] has been improperly used to generate arbitrary graphs [3–9, 13, 15–28, 30, 31]. In particular, the authors in [26] 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 [4–9, 13, 15–17, 19–21, 25, 27, 28, 30, 31].

Most of the authors using the generator in [11] claim that they group the graphs of their test bed by edge densities, but actually they group the graphs by threshold values. Our analysis of the instances introduced in [26] shows that the relation between the threshold d and the corresponding edge density δ is the following.

d	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
δ	0.00	0.02	0.08	0.18	0.32	0.50	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 Sect. 2.

4 Computational Results on Different Graph Classes

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 $X(n, \delta)$ we denote a set of ten instances with n vertices, bound $B = 150$, and conflict graph X with expected edge density $\delta \in \{0.02, 0.08, 0.18, 0.32, 0.50, 0.68, 0.82, 0.92, 0.98\}$ (the same densities of the instances used in [26]). In particular, we choose $n \in \{250, 1000\}$ and $X \in \{T, I, A\}$, where $X = T$ (I, A, respectively) when the conflict graph is a threshold (interval, arbitrary, respectively) graph.

The $T(250, \delta)$ and $T(1000, \delta)$ instances are exactly those in the classes 2 and 4 described in [26], 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.

As for the $I(n, \delta)$, the weight of the i-th vertex of the k-th instance coincides with the weight of the i-th vertex of the k-th instance of $T(n, \delta)$, and the arbitrary interval conflict graphs have been generated according to the interval graph generator in [2].[1]

As for the $A(n, \delta)$, the weight of the i-th vertex of the k-th instance coincides with 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 [28]: "*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 solved 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) defined in [4], 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. 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 introduced in [26], which are all solved to optimality within 50 min and with an average runtime of 2 min. For our analysis we solved the integer programming model with Cplex 12.6 on an Intel Core i7-3632QM 2.20 GHz with 16 GB RAM under a Linux operating system, setting a time limit of 600 s for each instance. The instances used in this section and many others can be downloaded at [1].

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 these latter are easier than those 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. The authors in [28] observe that the conflict graphs of the benchmark instances in [26] are interval graphs and not arbitrary graphs (actually they are not arbitrary interval ones). Nevertheless, to

[1] The generator in [2] 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.

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

		$n = 250$						$n = 1000$					
		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	–
	0.50	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

our knowledge, the authors in [28] 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 [3–9, 13, 15–28, 30, 31] and generated according to [11] are threshold graphs (and not arbitrary ones), and their edge density is not the declared one.

We also show that BPPC instances with threshold conflict graphs are computationally easier to solve than instances with interval or arbitrary conflict graphs (the instances used and many others are available at [1]). This behaviour confirms the behaviour of the computational complexity of many classical combinatorial problems on the three graph classes considered.

We believe that the reduced difficulty of these instances is mainly due to the structure of the conflict graph and not to "*the presence of capacity constraints on the cardinality of the color classes*" as suggested in [8]. This could be ascertained by solving BPPC instances with arbitrary interval conflict graphs and with arbitrary conflict graphs.

We also remark that the authors in [11] claim to use "the procedure described in" [29], but this is not true. In fact, the procedure in [29] generates "*edge* (i, j) *with probability*" $(p_i + p_j)/2$: by doing so it generalizes the uniform random graph generator and outputs arbitrary graphs.

References

1. Bacci, T.: http://www.iasi.cnr.it/~tbacci/ (2017)
2. Bacci, T., Nicoloso, S.: A heuristic algorithm for the bin packing problem with conflicts on interval graphs (2017). arXiv:1707.00496 [math.CO]
3. Basnet, C., Wilson, J.: Heuristics for determining the number of warehouses for storing non-compatible products. Int. Trans. Oper. Res. **12**, 527–538 (2005)
4. Brandão, F., Pedroso, J.P.: Bin packing and related problems: general arc-flow formulation with graph compression. Comput. Oper. Res. **69**, 56–67 (2016)
5. Capua, R., Frota, Y., Vidal, T., Ochi, L.S.: Um algoritmo heurìstico para o problema de bin packing com conflitos (2015). Manuscript http://www.din.uem.br/~ademir/sbpo/sbpo2015/pdf/146111.pdf
6. Capua, R., Frota, Y., Ochi, L.S., Vidal, T.: A study on exponential-size neighborhoods for the bin packing problem with conflicts. J. Heuristics **24**, 667–695 (2018)
7. Clautiaux, F., Khanafer, A., Hanafi, S., Talbi, E.: Le problème bi-objectif de bin-packing avec conflits (2011). Manuscript https://ori-nuxeo.univ-lille1.fr/nuxeo/site/esupversions/49ddd808-5557-41e2-bdfe-73b69e53264d
8. Cornaz, D., Furini, F., Malaguti, E.: Solving vertex coloring problems as maximum weight stable set problems. Discrete Appl. Math. **217**, 151–162 (2017)
9. Elhedhli, S., Li, L., Gzara, M., Naoum-Sawaya, J.: A branch-and-price algorithm for the bin packing problem with conflicts. INFORMS J. Comput. **23**, 404–415 (2011)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
11. Gendreau, M., Laporte, G., Semet, F.: Heuristics and lower bounds for the bin packing problem with conflicts. Comput. Oper. Res. **31**, 347–358 (2004)
12. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, Cambridge (1980)
13. Gschwind, T., Irnich, S.: Dual inequalities for stabilized column generation revisited. INFORMS J. Comput. **28**(1), 175–194 (2016)
14. Jansen, K., Oehring, S.: Approximation algorithms for time constrained scheduling. Inf. Comput. **132**(2), 85–108 (1997)
15. Joncour, C.: Problèmes de placement 2d et application à l'ordonnancement: modélisation par la théorie des graphes et approches de programmation mathématique. Ph.D. Thesis, Université de Bordeaux I (2010). Number 4173
16. Joncour, C., Michel, S., Sadykov, R., Sverdlov, D., Vanderbeck, F.: Column generation based primal heuristics. Electron. Notes Discrete Math. **36**, 695–702 (2010)
17. Jouida, S.B., Krichen, S.: A DSS based on optimizer tools and MTS meta-heuristic for the warehousing problem with conflicts. Inf. Process. Lett. **135**, 14–21 (2018)
18. Jouida, S.B., Ouni, A., Krichen, S.: A multi-start tabu search based algorithm for solving the warehousing problem with conflict. In: Thi, H.A.L., Dinh, T.P., Nguyen, N.T. (eds.) 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, pp. 117–128. Springer, Berlin (2015)
19. Khanafer, A.: Algorithmes pour des problèmes de bin packing mono- et multi-objectif. Ph.D. Thesis, Université Lille1 (2010). Number 40363
20. Khanafer, A., Clautiaux, F., Hanafi, S., Talbi, E.: The min-conflict packing problem. Comput. Oper. Res. **39**, 2122–2132 (2012)
21. Khanafer, A., Clautiaux, F., Talbi, E.: New lower bounds for bin packing problems with conflicts. Eur. J. Oper. Res. **206**(2), 281–288 (2010)
22. Khanafer, A., Clautiaux, F., Talbi, E.: Tree-decomposition based heuristics for the two-dimensional bin packing problem with conflicts. Comput. Oper. Res. **39**, 54–63 (2012)

23. Maiza, M., Guéret, C.: A new lower bound for bin packing problem with general conflicts graph. In: 1st Doctoriales STIC'09. Université de M'sila, Algérie (2009)
24. Maiza, M., Radjef, M.S.: Heuristics for solving the bin-packing problem with conflicts. Appl. Math. Sci. **5**, 1739–1752 (2011)
25. Muritiba, F.A.E.: Algorithms and models for combinatorial optimization problems. Ph.D. Thesis, Alma Mater Studiorum Università di Bologna (2010)
26. Muritiba, F.A.E., Iori, M., Malaguti, E., Toth, P.: Algorithms for the bin packing problem with conflicts. INFORMS J. Comput. **22**(3), 401–415 (2010)
27. Sadykov, R.: Modern branch-cut-and-price. Tech. Rep., Operations Research [cs.RO]. Universit de Bor-deaux (2019)
28. Sadykov, R., Vanderbeck, F.: Bin packing with conflicts: a generic Branch-and-Price algorithm. INFORMS J. Comput. **25**(2), 244–255 (2013)
29. Soriano, P., Gendreau, M.: Tabu search algorithms for the maximum clique problem. Dimacs Ser. Discrete Math. Theor. Comput. Sci. **26**, 221–236 (1996)
30. Wei, L., Luo, Z., Baldacci, R., Lim, A.: A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. INFORMS J. Comput. **32**(2), 428–443 (2020)
31. Yuan, Y., Li, Y., Wang, Y.: An improved aco algorithm for the bin packing problem with conflicts based on graph coloring model. In: 21th International Conference on Management Science & Engineering, Helsinki (2014)