



ELSEVIER



CrossMark

Procedia Computer Science

Volume 29, 2014, Pages 2192–2200

ICCS 2014. 14th International Conference on Computational Science



The Knapsack Problem with Three Practical Constraints

Rainne Florisbelo Gonçalves and Thiago Alves de Queiroz^{*}
Department of Mathematics, DM/CAC, UFG, 75704-020, Catalão-GO, Brazil
rainne.fg@gmail.com, taq@ufg.br

Abstract

This paper considers practical constraints for the unconstrained knapsack problem in its two-dimensional version, using integer programming. We first present an integer formulation for this knapsack problem, so couple constraints related with load balance, vertical (cargo) stability and fragility of the items also called load bearing. Using C language and the CPLEX solver, good results are obtained in an acceptable runtime considering instances from the literature.

Keywords: load balance, vertical stability, load bearing, unconstrained knapsack problem, integer programming

1 Introduction

The knapsack problem in its two-dimensional version consists of a rectangular bin B of dimensions (L, C) , L for length and C for width, and a set with n rectangular items each one with length l_i , width c_i and profit v_i . The aim is to pack a subset of items in B in order to maximize the total profit, such there is no overlapping between any pair of items and the resultant packing respects the bin dimensions. Under a perspective of cutting problems, we can cut the bin B in order to obtain such subset of items. In accordance with Garey and Johnson (1979), the knapsack problem is NP-hard. And, its two-dimensional version is strongly NP-hard.

The term *unconstrained* in knapsack problems means that there is unlimited copies of each item, so each item can be packed several times, with no limit. We deal with the two-dimensional unconstrained knapsack problem, denominated as 2UKP. Authors as Cintra et al. (2008) and Queiroz et al. (2012) presented exact algorithms based on dynamic programming to solve the two- and three-dimensional versions, respectively, however limited to the guillotine and staged cases. For the general case (non-guillotine), Birgin et al. (2012) presented effective heuristics based on five-order non-guillotine cuts and L-cuts. They solved several medium and large-sized instances to the optimality with these algorithms.

^{*} Corresponding author. Tel.: +55 64 3441-5316. Fax: +55 64 3441-5320.

The 2UKP appears in freight transportation industry when loading pallets in trucks, and in the cutting process when dealing with wood, glass and steel plates. It may also appear as subproblem of others combinatorial optimization problems like cutting stock and strip packing. In addition, we are interested in practical constraints that arise in freight transportation industry. Bischoff and Ratcliff (1995) enumerate some constraints and not so much has been done to consider them in packing problems. Bortfeldt and Wäscher (2013) presented a recent survey discussing several practical constraints used in the container loading problems, which can be viewed as a three-dimensional knapsack problem. In our study, we consider three practical constraints: load balancing, vertical (cargo) stability and items with fragility.

Load balance requires that the center of gravity of the packing must be in a safe region within the bin. It allows more stability when a vehicle performs a curve, and decreases the consumption of fuel in aircrafts. Kaluzny and Shaw (2009) presented integer programming models considering the load balance in aircraft transportation. Queiroz and Miyazawa (2013) also considered such constraint, but combined with multi-drop requirements for the strip packing problem. These last authors presented an integer model, valid inequalities and discussed about the size of the grid of points used for packing items.

Vertical (cargo) stability refers to a packing where no item can rotate (or fall down) if only the force of gravity acts on the items. It is essential in transportation, since if an item falls down it may damage itself and other ones. Some authors have been considered this constraint using a factor α for which the item's bottom face must have α percent supported by other items' upper face. Junqueira et al. (2012) solved the container loading problem with this assumption, and Gendreau et al. (2006) considered it in the capacitated vehicle loading problem with loading constraints. A realistic formulation uses physical concepts as in Silva et al. (2003).

Fragility, also referenced as load bearing constraint, refers to items that cannot support any other item, above and in direct contact with it. For non-fragile items there are no restriction about arrange items above or under them. In real cases, each item has a maximum load bearing strength that limits the maximum weight that such an item can bear from above it. Queiroz and Miyazawa (2013) proposed an effective procedure based in physics concepts to deal with the load bearing constraint. It was embedded in a branch-and-cut algorithm, which solved the strip packing problem. Junqueira et al. (2012) considered such constraint in an integer formulation for the container loading problem. They assumed that each item has a maximum admissible pressure that it can bear at any point of its upper face, and any of these points support the same pressure.

In this paper we discuss an integer formulation for the 2UKP in Section 2, and model load balance, vertical stability and fragility constraints as integer expressions in Section 3. We adapt from Junqueira et al. (2012) the expressions for the load bearing and vertical stability constraints. The resultant integer formulations are solved with the IBM ILOG CPLEX solver in which numerical experiments, in Section 4, using instances from the specialized literature validate them. Optimal solutions are computed in an acceptable time for practical applications. Finally, Section 5 shows the conclusions and perspectives for future research.

2 Problem Formulation

The mathematical formulation for the 2UKP is based in Beasley (1985) formulation. This author developed an integer formulation solved with an exact procedure over a tree search that considers items packed at the discretization points of Herz (1972). The discretization points are obtained as an integer conic combination of the items length (width) for the length-direction (for the width-direction). We assume that the bin starts at $(0, 0)$ in Cartesian plane, each item is packed by its bottom-left corner, and the input data of any instance are all integers. Moreover, Herz (1972) stated that the discretization points are sufficed to obtain the optimal solution.

Let P be the set of discretization points of the length and Q of the width. So, formulation (1)-(3) is capable of solve the 2UKP, in which variable $x_{ipq} = 1$ means that item i is packed at point $(p, q) \in P \times Q$. Note that we assume the bin discretized on a grid of points given by $P \times Q$. We consider the set R_{ipq} as that containing the points (r, s) of the grid where item i packed at any of them covers (p, q) .

$$\begin{aligned} & \text{Maximize } \sum_{i=1}^n \sum_{p \in P} \sum_{q \in Q} v_i x_{ipq} & (1) \\ & \text{Subject to: } \sum_{i=1}^n \sum_{(r,s) \in R_{ipq}} x_{irs} \leq 1, \quad \forall p \in P, \forall q \in Q & (2) \\ & \quad x_{ipq} \in \{0, 1\}, \quad i = 1, \dots, n, \forall p \in P, \forall q \in Q & (3) \end{aligned}$$

The objective function (1) aims to maximize the total value/profit packed in the bin. Constraints (2) avoid items overlapping, so each point (p, q) of the grid must be covered by at most one item packed at some point (r, s) , while constraints (3) impose that all variables are binary. Note that any item can be packed “unlimited” times, contrary to the constrained version in which a constraint like $lb_i \leq \sum_{p \in P} \sum_{q \in Q} x_{ipq} \leq ub_i$ is used for each item i in order to limit the minimum (lb_i) and maximum (ub_i) number of copies allowed in the solution.

Scheithauer and Terno (1996) presented new sets of points, called reduced raster points. These sets are derived from the discretization points, however there is no guarantee about the optimality when using them, that is, perhaps the optimal solution is lost. It is still open in the literature whether with the reduced raster points there is (no) loss of generality. Based on computational results from Birgin et al. (2012) and Queiroz et al. (2012), we believe that there is no loss of generality when using the reduced raster points, so we consider them instead of the discretization points. Equation (4) shows how to compute the set of reduced raster points of the length from the respective set P . The same holds for the width.

$$\bar{P} = \{ \langle L - s \rangle \mid s \in P \}, \text{ where } \langle s \rangle = \max\{t \in P \mid t \leq s\} \tag{4}$$

Cintra et al. (2008) presented a dynamic programming algorithm to obtain the sets of discretization points, then follows from eq. (4) the respective sets of reduced raster points.

3 Practical Constraints

3.1 Load Balance

In the load balance constraint, the resultant packing must has its center of gravity within a region of the bin. This region is called envelope (Kaluzny and Shaw, 2009) and it is determined considering safe and economic conditions for the packing. In some situations it consists of the geometrical mid-point of the bin. Others researches relax it to be within an interval in length- and width-directions, that is a rectangular envelope defined by $\{[x_{start}, x_{end}], [y_{start}, y_{end}]\}$.

In order to calculate the center of gravity, each item i has a mass and is considered a rigid body. Without loss of generality, we consider the mass proportional to the item’s area, that is, $m_i = l_i c_i$, and near the surface of the earth the center of gravity coincides with the center of mass, assuming a uniform gravity field. The center of mass can be viewed as the mean location of distribution of mass for a body and is given in function of coordinates (CM_x, CM_y) (Hibbeler, 2010). For an optimal

solution, we require that these coordinates must be in a given envelope, resulting in the following constraints for the length-direction:

$$\frac{\sum_{i=1}^n \sum_{p \in \bar{P}} \sum_{q \in \bar{Q}} m_i \left(p + \frac{l_i}{2}\right) x_{ipq}}{\sum_{i=1}^n m_i} \geq x_{start} \tag{5}$$

$$\frac{\sum_{i=1}^n \sum_{p \in \bar{P}} \sum_{q \in \bar{Q}} m_i \left(p + \frac{l_i}{2}\right) x_{ipq}}{\sum_{i=1}^n m_i} \leq x_{end} \tag{6}$$

Constraints (5) and (6) impose that the x -coordinate, related with the length-direction, of the center of mass of the final packing must be between $[x_{start}, x_{end}]$. Similarly, we derive constraints for the y -coordinate, related with the width-direction, substituting l_i by c_i , p by q , and $[x_{start}, x_{end}]$ by $[y_{start}, y_{end}]$.

3.2 Vertical Stability

We consider the vertical stability for the case in which a factor α is used to assure that each item has its bottom face α percent supported by other items immediately under and in direct contact with it. For $\alpha = 100\%$, it is required that each item must be fully supported, while $\alpha = 0\%$ does not impose any restriction so items may float in mid-air. The α is called vertical stability parameter and ranges between 0% and 100%.

As observed by Queiroz and Miyazawa (2013), the value of α does not solve the problem of vertical stability at all, since an item i is considered stable if its center of mass lies on items under and in direct contact with i , or between two of these items, or on the floor of the bin, otherwise i can rotate and fall down. With a factor α we still get good solutions for practical applications, especially when $\alpha = 100\%$, although solutions vertically stable may be lost for the general case, that is, without considering this factor. Constraints in (7) define the vertical stability in accordance with Junqueira et al. (2012).

$$\sum_{\{i=1, \dots, n \mid s-c_i \geq 0\}} \sum_{\{p \in \bar{P} \mid r-l_i+1 \leq p \leq r+l_j-1\}} L_{ij} x_{ip(s-c_i)} \geq \alpha l_j x_{jr_s} \tag{7}$$

$j = 1, \dots, n; \forall r \in \bar{P}; \forall s \in \bar{Q}$

Function $L_{ij} = \min(p + l_i, r + l_j) - \max(p, r)$ defines the first point in the upper face of i that supports j . Constraints in (7) impose that each item j , which is not directly supported by the floor, must have α percent of its bottom face supported by other items under and in direct contact with it. Note that the term “face” (bottom or upper) stands for the length of the item, since we are dealing with the two-dimensional version and assume that all items have the same depth (for example, equal to one).

3.3 Load Bearing

In accordance with Junqueira et al. (2012), each item i has a maximum number of items that can be arranged above it so exerting some pressure on its upper face. In other words, item i has a maximum value of pressure σ_i that its upper face supports without be damaged. In this case, we assume that each point of its upper face can support this value. Moreover, the pressure exerted by an item is uniformly distributed throughout its bottom face, so we assume it given by $\mathbb{P}_i = \frac{m_i g}{l_i}$, where $g = 9.8$ is the gravitational acceleration. With this in mind, eq. (8) presents load bearing constraints.

$$\forall r \in \bar{P}; \forall s \in \bar{Q} \quad \sum_{j=1}^n \sum_{\{p \in \bar{P} | r-l_j+1 \leq p \leq r\}} \sum_{\{q \in \bar{Q} | s+1 \leq q \leq c-c_j\}} \mathbb{P}_j x_{jpq} \leq \sum_{i=1}^n \sum_{\{p \in \bar{P} | r-l_i+1 \leq p \leq r\}} \sum_{\{q \in \bar{Q} | s-h_i+1 \leq q \leq s\}} \sigma_i x_{ipq} \tag{8}$$

Observing constraints in (8), the right side checks for an item i in which the point (r, s) is at its upper face. If true, then all items j , arranged above i , that exert pressure on it must respect its maximum admissible pressure σ_i . As commented σ_i is assumed equal for each point of the upper face of i .

Queiroz and Miyazawa (2013) proposed an effective method based on physics concepts which checks if a packing respects the load bearing constraints. However, their method cannot be embedded in an integer formulation due to the large number of constraints that it raises up.

4 Computational Experiments

We use instances of two-dimensional packing problems for purpose of numerical tests. These instances are from Beasley (1985), called GCUT1-12, Morabito et al. (1992), called M1-5, and Hifi (2001), called MW1-5, totaling 22 instances.

The integer formulation and the constraints were codified in the C language programming and solved with the ILOG CPLEX 12 Callable Library (IBM, 2009). We set a time limit of 3600 seconds for each instance, and use a computer with Intel Xeon 2.4 GHz processor, 8 GB of RAM memory and Linux operating system.

The CPLEX solver provides a framework to solve integer linear models with a branch-and-cut algorithm. Some cuts inserted by CPLEX are clique, cover, disjunctive, flow, Gomory and zero-half cuts. Besides that, there are a set of parameters that can be modified in order to speed up the resolution of the model. We use the default values for these parameters in our experiments.

Preliminary tests are presented in order to compare the integer formulation that solves the 2UKP with heuristic algorithms from the literature. We consider solutions obtained with the Five-block and L-approach algorithms of Birgin et al. (2012). Table 1 exhibits the results of the integer formulation in columns 3 and 4, while column 1-3 show the instance’s name, bin dimensions and number of items, respectively. The remained columns present the results from Birgin et al. (2012), where they used a computer with 2.4 GHz Intel Core 2 Quad Q6600 processor, 8 GB of RAM memory and Linux Operating System.

Observing Table 1, with the integer formulation, we compute optimal solution for 20 out of 22 instances, where only two ones are not solved due to lack of memory, GCUT4 and GCUT8. The large number of points on the grid can explain it, since the number of variables and non-overlapping constraints increase accordingly to. On the other hand, we improve the solution of M1 compared to Five-block. For all other instances, the same value of solution is obtained comparing the three approaches. It is worth mentioning that Birgin et al. (2012) used a reduction procedure that eliminates dominated items of the instance, so the number of patterns to be considered is eventually smaller than those of the original instance. However, such procedure cannot be used if practical constraints are considered in the model.

Comparing the computation time, with the integer formulation we spent 25.34 seconds, on average, for each instance, while the Five-block and L-approach required 0.29 and 8.09 seconds, respectively. Although the resolution of integer models is not so fast compared with heuristic algorithms, the worst computation time is of 231.92 seconds for the instance GCUT12, which is

considered acceptable in practical situations. Moreover, heuristic algorithms may not return optimal solutions. More details can be found in Table 1.

Birgin et al. (2012)								
Instance	Bin dimensions (L, C)	Number of items n	Solution	Time (s)	<i>Five-block</i>		<i>L-approach</i>	
					Solution	Time (s)	Solution	Time (s)
GCUT1	(250, 250)	10	58 480	0.02	58 480	< 0.00	58 480	0.72
GCUT2	(250, 250)	20	61 146	2.01	61 146	< 0.00	61 146	0.96
GCUT3	(250, 250)	30	61 275	29.41	61 275	0.02	61 275	4.21
GCUT4	(250, 250)	50	-	-	61 918	0.06	61 918	2.22
GCUT5	(500, 500)	10	246 000	0.03	246 000	< 0.00	246 000	0.67
GCUT6	(500, 500)	20	243 598	0.42	243 598	< 0.00	243 598	0.74
GCUT7	(500, 500)	30	244 306	0.93	244 306	< 0.00	244 306	0.92
GCUT8	(500, 500)	50	-	-	247 815	0.10	247 815	32.57
GCUT9	(1000, 1000)	10	971 100	0.01	971 100	< 0.00	971 100	0.68
GCUT10	(1000, 1000)	20	982 025	0.37	982 025	< 0.00	982 025	0.76
GCUT11	(1000, 1000)	30	980 096	6.14	980 096	< 0.00	980 096	1.92
GCUT12	(1000, 1000)	50	979 986	231.92	979 986	0.06	979 986	16.95
M1	(100, 156)	10	15 073	4.51	15 054	< 0.00	15 073	1.13
M2	(253, 294)	10	73 255	96.39	73 255	0.04	73 255	7.90
M3	(318, 473)	10	147 386	1.21	147 386	< 0.00	147 386	1.10
M4	(501, 556)	10	266 233	0.44	266 233	< 0.00	266 233	0.76
M5	(750, 806)	10	579 883	0.88	579 883	< 0.00	579 883	1.08
MW1	(100, 156)	10	3 882	0.81	3 882	< 0.00	3 882	1.22
MW2	(253, 294)	10	24 950	8.84	24 950	0.01	24 950	6.17
MW3	(318, 473)	10	37 068	1.26	37 068	< 0.00	37 068	1.01
MW4	(501, 556)	10	59 576	6.19	59 576	< 0.00	59 576	1.07
MW5	(750, 806)	10	189 924	0.74	189 924	< 0.00	189 924	0.94

Table 1: Results for the 2UKP.

For those constraints discussed in Section 3, we consider separately each one in the integer formulation of the 2UKP. As Birgin et. al (2012) did not consider those constraints, we compare the results obtained between themselves, looking at how the final pack may be improved with each one. For the load balance constraint, we assume the envelope given by $\{[0.3L, 0.7L], [0.3C, 0.7C]\}$, that is, the resultant center of mass must be between 30% and 70% of the length and width. For the vertical stability, we consider $\alpha = 100\%$, so each item must has its bottom face fully supported on other items or on the bin's floor, and in the load bearing, we assume $\sigma_i = l_i c_i$. All these assumptions are in accordance with the specialized literature.

Table 2 shows the results obtained for the load balance and load bearing constraints. It is worth to mention that no solutions are reported for the vertical stability constraint. Observing eq. (7), note that item i is packed at point (r, s) only if exists items packed along the width-direction at coordinate $s - c_i$, but subtraction operation is not closed in the reduced raster points set, such it is performed by using the operation $\langle s - c_i \rangle$. When using operator $\langle \rangle$, items may float in mid-air, since points of the grid are disregarded, contrary to the stability condition which requires that items must be in direct contact

with another ones. This problem increases accordingly the value of α increases too. A detailed discussion about the reduced raster points is performed in Scheithauer (1997), as well as Birgin et al. (2012) discussed that the number of patterns (and points of the grid) may reduce in almost 50% compared to the discretization points.

Instance	Load Balance		Load Bearing	
	Solution	Time (s)	Solution	Time (s)
GCUT1	0	0.02	56 460	0.03
GCUT2	0	0.12	60 536	26.96
GCUT3	0	1.4	61 036	364.27
GCUT4	-	-	-	-
GCUT5	0	0.04	246 000	0.23
GCUT6	0	0.02	238 998	4.47
GCUT7	0	0.16	242 567	24.85
GCUT8	-	-	-	-
GCUT9	0	0.02	971 100	0.10
GCUT10	0	0.04	982 025	2.98
GCUT11	0	0.56	980 096	73.46
GCUT12	0	7.19	979 986	2347.65
M1	15 073	8.83	15 024	15.04
M2	72 708	106.32	72 708	246.55
M3	147 386	1.69	142 817	23.63
M4	0	0.27	265 768	2.03
M5	612 436	1.37	612 436	4.04
MW1	3 882	1.14	3 882	4.22
MW2	24 950	12.99	24 950	48.47
MW3	37 068	1.33	37 068	9.78
MW4	59 576	2.15	59 576	6.98
MW5	189 924	1.25	189 924	3.95

Table 2: Results for the Practical Constraints.

The results in Table 2 show that for the load balance constraint, only 9 out of 22 instances are solved considering the envelope mentioned before. In instances GCUT1-12 and M4, for each one the solution value is “0” means that no feasible solution exist for the envelope under consideration. Of course, if we relax the size of the envelope, feasible solutions can be obtained for more instances, although the reduced raster points may difficult to get such solutions. On the other hand, the computation time decreases (corresponds to 7.35 seconds on average) if we compare with Table 1, the case without practical constraints. Note that the longest time occurred for M2, requiring 106.32 seconds.

For the load bearing constraint, 20 instances are solved to the optimality spending 160.48 seconds on average. Instance GCUT12 is the hardest one requiring 2347.65 seconds. Contrary to the load balance, the load bearing requests more computation time compared with Table 1 for all the instances. Moreover, the value of the solution decreases for some instances compared with the case without practical constraints.

5 Concluding Remarks

In this paper we present an integer formulation based on a grid of reduced raster points for the two-dimensional unconstrained knapsack problem. Besides that, three practical constraints of freight transportation are taken in consideration, namely load balance, vertical stability and load bearing constraints. All these constraints are presented as integer inequalities and embedded in the integer formulation.

The integer formulation is competitive compared with heuristics from the literature, computing optimal solutions for 20 out of 22 instances. On the other hand, some of the constraints under consideration do not work properly with the set of reduced raster points, that are vertical stability and load balance constraints. In such cases, it is necessary to consider the discretization points as performed by Junqueira et al. (2012). Moreover, add practical constraints in the original problem increase its difficulty and the solution may be the worst in terms of value and occupation of the bin.

Future works will consider new practical constraints, as multi-drop requirements, items spacing, orthogonal rotation of the items, etc. Also to consider these constraints in the three dimensional version, as well as the combination of these constraints to deal with realistic scenarios appearing in freight transportation.

Acknowledgements. The authors thank CNPq and FAPEG for the financial support.

References

- Beasley, J.E. (1985). Algorithms for unconstrained two-dimensional guillotine cutting. *Journal of the Operational Research Society*, 36(4): 297-306.
- Birgin, E.G.; Lobato, R.D.; Morabito, R. (2010). An effective recursive partitioning approach for the packing of identical rectangles in a rectangle. *Journal of the Operational Research Society*, 61: 306-320.
- Bischoff, E.E.; Ratcliff, M.S.W. (1995). Issues in the development of approaches to container loading. *Omega*, 23(4): 377-390.
- Bortfeldt, A.; Wäscher, G. (2013). Constraints in container loading – a state-of-the-art review. *European Journal of Operational Research*, 229: 1-20.
- Cintra, G.F.; Miyazawa, F.K.; Wakabayashi, Y.; Xavier, E.C. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191: 59-83.
- Garey, M.R.; Johnson, D.S. (1979). *Computer and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.
- Gendreau, M.; Iori, M.; Laporte, G.; Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3): 342–350.
- Herz, J.C. (1972). A recursive computational procedure for two-dimensional stock-cutting. *IBM Journal of Research Development*, 1: 462-469.
- Hibbeler, R.C. (2010). *Statics and Mechanics of Materials*. 3^a ed. Prentice Hall.
- Hifi, M. (2001). Exact algorithms for large-scale unconstrained two and three staged cutting problems. *Computational Optimization and Applications*, 18: 63-88.

- IBM. (2009). ILOG[®] CPLEX[®] V12.1 – *User's Manual for CPLEX[®]*. IBM Corporation.
- Junqueira, L.; Morabito, R.; Yamashita, D.S. (2012). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers and Operations Research*, 39: 74-85.
- Kalzuny, B.L.; Shaw, R.H.A.D. (2009). Optimal aircraft load balancing. *International Transactions in Operation Research*, 16(6): 767-787.
- Morabito, R.; Arenales, M.; Arcaro, V.F. (1992). An and-or-graph approach for two-dimensional cutting problems. *European Journal of Operational Research*, 58: 263-271.
- Queiroz, T.A.; Miyazawa, F.K. (2013). An Integer Programming Model for the Two-dimensional Strip Packing Problem with Multi-drop and Load Balancing Constraints. *International Journal of Production Economics: Special Issue on Cutting and Packing*, 145 (2): 511-530.
- Queiroz, T.A.; Miyazawa, F.K.; Wakabayashi, Y.; Xavier, E.C. (2012). Algorithms for 3D guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing. *Computers & Operations Research*, 39 (2): 200-212.
- Scheithauer, G. (1997). Equivalence and dominance for problems of optimal packing of rectangles. *Ricerca Operativa*, 27: 3-34.
- Scheithauer, G.; Terno, J. (1996). The G4-heuristic for the pallet loading problem. *Journal of the Operational Research Society*, 47: 511-522.
- Silva, J.L.C.; Soma, N.Y.; Maculan, N. (2003). A greedy search for the three-dimensional bin packing problem: the packing static stability case. *International Transactions in Operational Research*, 10(2): 141-153.