www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 12(2), May-Jun 2021, 16-24. ISSN: 2007-1558.

Analysis of the Relationship Between Characteristics and Hardness of Strip Packing Problem Instances

Edgar Oviedo-Salas[1], J. David Terán-Villanueva[1], Salvador Ibarra-Martínez[1], José Antonio Castan-Rocha[1], Mirna Patricia Ponce-Flores[2], Julio Laria-Menchaca[1], Mayra Guadalupe Treviño-Berrones[1]

[1]Universidad Autónoma de Tamaulipas (UAT),
Centro Universitario Tampico Madero S/N,
Universidad Poniente, Tampico, Tamps. 89109, México.

[2]Tecnológico Nacional de México,
Instituto Tecnológico de Ciudad Madero (ITCM),
Juventino Rosas 206, Los Mangos, 89440 Cd Madero, Tamps.

eaos9407@gmail.com, david_teran01@yahoo.com.mx, sibarram@docentes.uat.edu.mx, jacastan@docentes.uat.edu.mx, mirna_poncef@hotmail.com, jlaria@docentes.uat.edu.mx, mgtrevino@docentes.uat.edu.mx

Abstract. *The Two-Dimensional Strip Packing Problem (SPP) is an NP-Hard problem where a list of rectangular objects must be accommodated in a container to minimize the objects' total height while avoiding overlapping. It is common to use meta-heuristic algorithms to solve this problem, which produces near-optimal solutions in a reasonable time. In this paper, we analyze several characteristics of SPP instances, looking for features that define their hardness. For further development of new algorithms that make use of this knowledge.*

Keywords: Strip Packing Problem, Characterization, Instance Analysis, Combinatorial problem.

Article Info
Received July 12, 2020
Accepted April 15, 2021

1 Introduction

The arrangement of two-dimensional objects, called strip packing problem, is commonly found in the industrial area, where it is most commonly used to cut or arrange parts in materials such as paper, cloth, wood, glass, and other materials. Reducing the height used to accommodate as many as possible inside these containers.

The Strip Packing Problem (*SPP*) is an NP-Hard optimization problem [1], for which the use of heuristic or meta-heuristic algorithms is required. The strip packing problem is defined as follows, let R be a rectangular container with a fixed width W_r and an infinite height, the problem is to accommodate a set of rectangular objects $O = \{o_1, o_2, \ldots, o_n\}$ within R in order to minimize the height used.

Gaticia et al. approached this problem in [2]; they implemented an algorithm called *Strip Packing Problem Game (SPPG)* using players, patterns, data mining, heuristics, and decision trees. The algorithm analyzes a player's moves, trying to solve a puzzle representing a *Strip Packing Problem* instance, through data mining techniques and patterns. Their experimental results showed that *(SPPG)* could identify game patterns from previous plays.

In [3], Zhang et al. proposed an improved hybrid meta-heuristic algorithm of variable neighbourhood search called Hybrid Algorithm (*HA*), based on block pattern construction. Their algorithm has three phases. First, it uses the less waste strategy, consisting of scoring rules to take the objects that fit with less waste into the strip. The second phase selects a better sorting sequence to finish an initial solution. Finally, the third phase constructs a different neighbourhood; based on block patterns. Their results show that *HA* is efficient to select a neighbour dynamically. Furthermore, computational tests demonstrate that *HA* surpasses other literature approaches for hard instances of the *Strip Packing problem*.

In [4], Wei et al. analyzed the SPP with unloading constraints; this means that the objects are arranged in a two-dimensional space for transport and unload- ing. However, the items belong to different customers, and the accommodation impacts the cost of the solution. Thus, this problem aims to minimize the total height while the packaging must satisfy the discharge conditions. This proposal uses segment trees, heuristics for open spaces, and random local searches. The results showed that their implementation outperforms the literature; while testing 283 instances of the two-dimensional orthogonal packing problem (*2lcvrp, Chr, Brk, Ben, Htu, Hop, Bea*).

In [5], Martin et al. approached the Constrained Two-dimensional Guillotine Placement Problem (C2GPP), which is mainly different from the SPP in restricting the type of cut allowed. This problem only allows orthogonal cuts, which means that the cuts must be parallels to the edges of the objects, and the cut produces two rectangles. The authors proposed implementing a nonlinear integer function to obtain linear programming for nonlinear models and decision trees. They developed constraints for each model, such as the step cut patterns. The tests were carried out on three data sets (*Chr, Bea*); the study concludes that models based on ascending storage lead to optimal or semi-optimal solutions with a reasonable computational cost.

For this problem, as for many others, there are differences in the hardness of the instances. In this research, we aim to identify the main characteristics of the instances that define their hardness. Thus, we analyze several data sets used in [4], Wei et al. *(2lcvrp, Chr, Brk, Ben, Htu, Hop, Bea)* and [5], Martin et al. *(Chr, Bea)* and apply data mining techniques to identify and classify the hardness of the instances and the characteristics that make them hard.

The remainder of the paper is as follows, Section 2 specifies the set of instances used in this work, Section 3 shows the methodology used, Section 4 contains the experimental results. Finally, Section 5 presents conclusions and future work.

2 Instances and their classification

Table 1, describes the used instances *2lcvrp, Chr, Brk, Ben, Htu, Hop, Bea*. Where the first column shows the name of the data set, the second column contains the cite that uses it, while the third and fourth columns show the number of instances and the range of the number of objects for each data set, respectively.

Table 1. Used Instances

Data Set	Author	Inst	n
2lcvrp	Gendreau et al [6]	180	$15-255$
Chr	Christofides and Whitlock [7]	3	$10-70$
Brk	Burke et al [8]	12	$10-500$
Ben	Bengtsson [9]	10	$20-200$
Htu	Hopper and Turton [10]	9	$16-28$
Hop	Hopper and Turton [11]	14	$17-199$
Bea	Beasley et al [12-13]	25	$10-22$

As we aim to study the characteristics of the instances that make them hard, we need to identify the hard to solve instances. Therefore, we solve every instance twenty times using a GRASP algorithm, not described in this paper, to produce an average error for each instance. We incline to obtain the average error in this way because the state of the art papers did not show results for individual instances, instead they showed their average error for every data set.

Once we have the average error for each instance, we carried out a k-means clustering algorithm [14] to group the instances according to their hardness. This algorithm makes groups of sets of objects into k clusters and uses a distance function to measure their proximity. One of the main drawbacks of these algorithms is that it needs a parameter the number of clusters desired, which is not always clear to the researcher. One way to tackle this problem is to use the elbow method [15] to find the right number of groups to use in the k-means algorithm.

Figure 1. Elbow Graph

As we can see in Figure 1, the *(elbow technique)* suggests using k = 5, because that value is where the curve lowers its error reduction. Figure 2 shows the result of the k-means algorithm for k = 5.

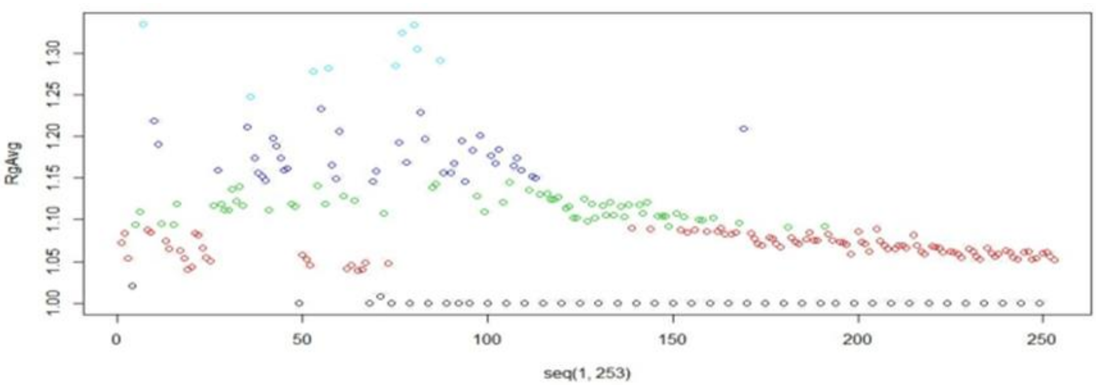

Figure 2. Classification with 5 Clusters.

Thus we named each group: extremely easy (*ee*), easy (*e*), normal (*n*), hard (*h*), and very hard (*vh*). Table 2 shows the number of instances in each group, where the first column shows the group's name, and the second column shows the total number of instances in the group.

Table 2. Groups Assigned

Cluster	Total
Extremely Easy	41
Easy	101
Normal	63
Hard	39
Very Hard	9

For each one of these groups, we took five random instances for further analysis. Table 3 shows the cite, the name of the data set, the number of instances selected from each data set, and the range of size of the instances in the data set for columns one to four respectively.

Table 3. Selected Instance

Cite	Data Set	Inst	n
Gendreau et al [7]	2lcvrp	12	$15-255$
Burke et al [9]	Brk	6	$10-500$
Hopper and Turton [11]	Htu	1	25
Hopper and Turton [12]	Hop	3	$25-199$
Beasley et al [13-14]	Bea	3	$30-50$

Table 4 shows the 25 selected instances, classified according to the averages of the error range obtained. Where the first column shows the name of the data set, the second column contains the instance name, while the third and fourth columns show the group member of the instance and the error for each instance, respectively.

It is worth mentioning that the easy group (*e*) is the one with the most elements with 121 instances, followed by the normal, extremely easy, hard, and very hard groups (*n*) with 63, 41, 39, and 9 instances, respectively.

Finally, once we obtained the five data sets classification per group, we calculate the percentage of objects they represent, their average width, the average height, and each group's standard deviation for selected instances.

3 Instances Object Analysis

After the selection of the five instances for each group, we analyze the characteristics of the objects of the instances. Thus, for each instance, we calculate the ratio of the width of the objects regarding the width of the strip (see eq.1),

Table 4. Instances Results

Data Set	Name	Difficult	Error
2lcvrp	2lcvrp112	*ee*	1
2lcvrp	2lcvrp117	*ee*	1
2lcvrp	2lcvrp141	*ee*	1
Brk	Brk01	*ee*	1.02
Bea	Bea23	*ee*	1.008
Brk	Brk11	*e*	1.06
2lcvrp	2lcvrp122	*e*	1.07
2lcvrp	2lcvrp129	*e*	1.07
2lcvrp	2lcvrp167	*e*	1.06
2lcvrp	2lcvrp177	*e*	1.06
Brk	Brk12	*n*	1.09
Htu	Htu06	*n*	1.13
Hop	Hop14	*n*	1.11
Bea	Bea08	*n*	1.11

2lcvrp	2lcvrp118	*n*	1.09
Brk	Brk07	*h*	1.21
Brk	Brk08	*h*	1.19
Hop	Hop06	*h*	1.14
Bea	Bea12	*h*	1.2
2lcvrp	2lcvrp96	*h*	1.2
Brk	Brk04	*vh*	1.21
Htu	Htu02	*vh*	1.24
2lcvrp	2lcvrp14	*vh*	1.29
2lcvrp	2lcvrp07	*vh*	1.33
2lcvrp	2lcvrp08	*vh*	1.3

$$width\_ratio_i = (ow_i/CW) \qquad 1 \leq i \leq |O| \quad (1)$$

Where:

- ow_i is the width of the object i.
- CW is the container width.

The ratio of the objects' height regarding the *best solution height* found (see eq.).

Where:

- oh_i is the height of the object i.
- BSH is the *best solution height*.

For every instance of Table 1, we used the k-means algorithm and the elbow technique to determine the proper number of clusters, regarding the height and $width\_ratio$.

However, it is worth noting that the elbow technique suggested the same number of clusters for all the studied instances. Subsections 3.1 and 3.2 present two instances and the methodology to produce the clustering on their objects.

4 Experimental Results

In this section we present several graphs of the percentage of the objects (y axis) of the sampled instances of groups (ee, e, n, h, vh), and the average $width\_ratio/height\_ratio$ as percentage (x $axis$). Tables 5 and 6 show an example of these data.

4.1. Graphical analysis for ee

Figures 7 and 8 show the sampled instances' graphical data, from the ee set, regarding the objects' width and height respectively. Here, we can see that the 2lcvrp1xx instances were designed to generate a satisfying filling of the container because all the objects have 100% of $width\_ratio$ and have one unit of height, which is confirmed in Figure8. However, those values are below and around one percentage because their best solution height is 75, 134, and 255, and by applying Eq. 1 and 2 those values will present such behaviour. Additionally, the rest of the instances in ee are designed to produce a width near 100%; e.g., if we take three elements from 26% (Bea23) plus the other three elements from the 7%, it will reach 99% of the width of the container.

Furthermore, regarding Figure 8, the objects around 46% (Bea23) and 52% are near half 100%, which makes them ideal to complement each other, and the objects of 1% fill the remaining space. It is important to highlight that these values are mean, which produce worse matches than real individual values.

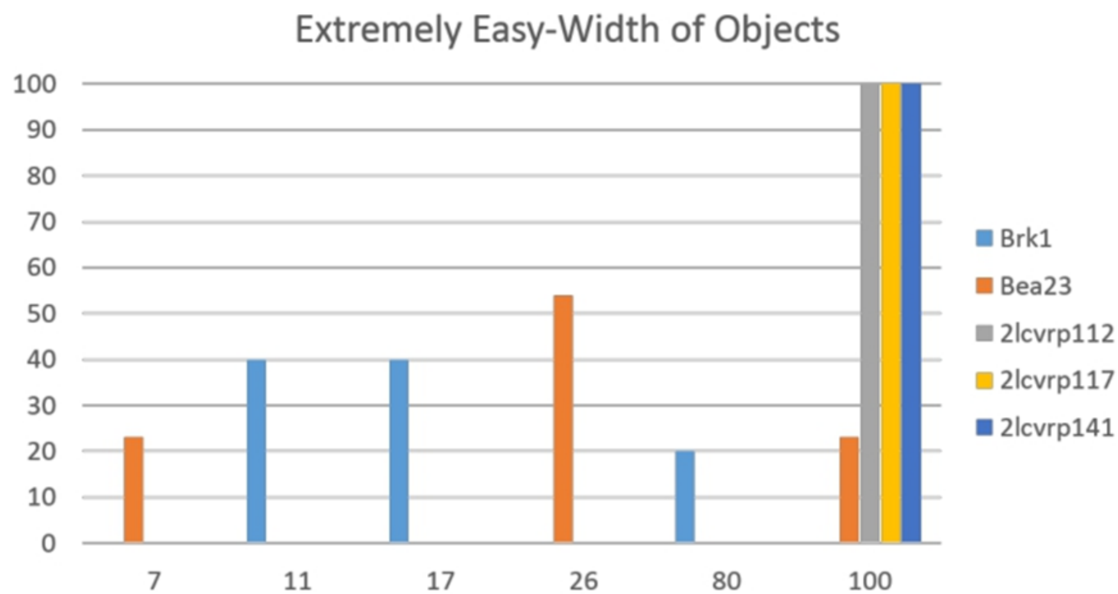

Figure 7. Width of Objects Extremely Easy Group

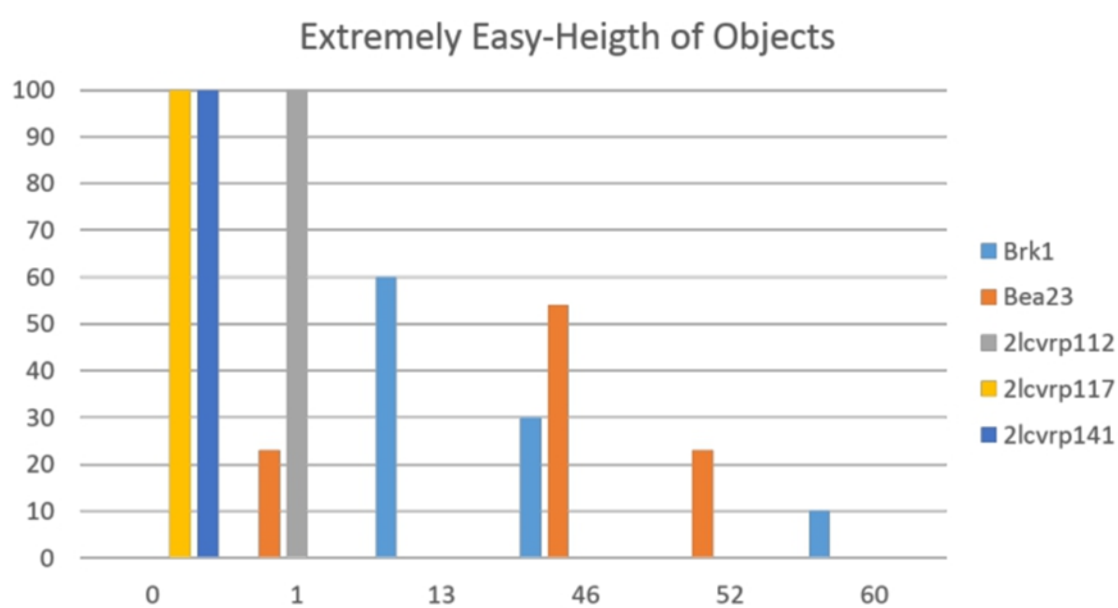

Figure 8. Height of Objects Extremely Easy Group

4.2. Graphical analysis for *e*

Figures 9 and 10 describe the result obtained from the sampled instances of the set *e*. Here we can see that *Brk11* is the easiest instance because of the large number of small size objects, which help fill empty spaces.

The rest of the instances hold the same amount of objects, regarding the width, for sizes 8-10, 21-25, and 34-42. On the other hand, a large number of objects have a low height, except for one instance, which has a percentage of the height of 36 and 49. However, the objects in this instance can be tiled, and for the remaining space, we can use objects of 5%. Figure 9 shows that *small* objects are predominant, with roughly 40% of the objects, while Figure 10 shows the same predominance of *small* objects with 59.9% of the objects.

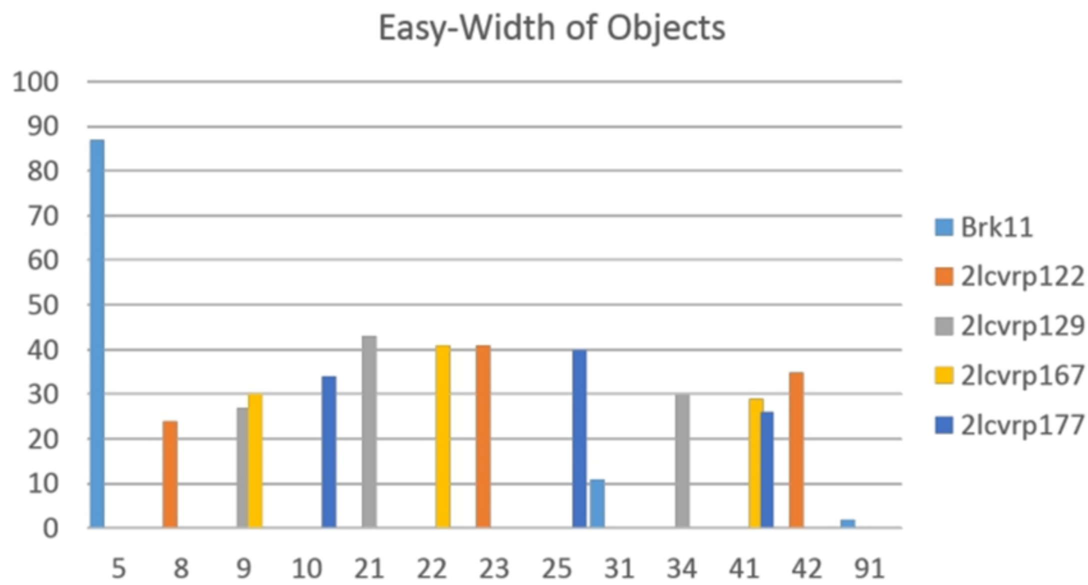

Figure 9. Width of Objects Easy Group.

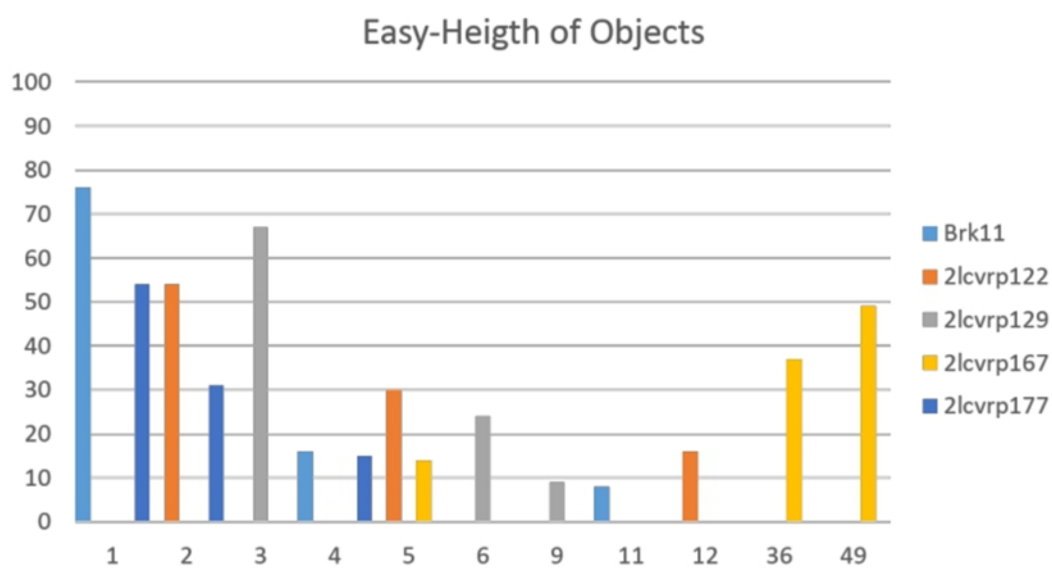

Figure 10. Height of Objects Easy Group.

5 Conclusions

In this research, we used datasets from state of the art to identify instances of hardness for the two-dimensional strip packing problem.

These datasets were processed through the use of clustering techniques, which, according to the results obtained, allowed the classification of five groups named as; extremely easy (*ee*), easy (*e*), normal (*n*), hard (*h*), and very hard (*vh*).

Once we obtained the groups, we sample five instances of each group to further analyse their objects' width and height.

These analysis, allow us to note that the number of objects in the instances was not relevant, neither the $width\_ratio$ nor the distribution of objects in the groups *small*, *medium*, and *large* regarding the $width\ ratio$ and $height\_ratio$. However, the $height\_ratio$ of the objects does show a direct relation with the hardness of the instances. It is not that the number of objects in the *large* group was higher; rather, the heights of the objects (regarding the *best solution height*) for all the groups of objects (*small*, *medium*, and *large*) were taller.

Therefore, as future work, we propose using the analysis of the $height\_ratio$ as a way to identify the hardness of the instance and therefore produce an ad-hoc algorithm to solve each kind of instance or create a hyper-heuristic algorithm.

6 References

[1] Baker, B. S., Coffman, Jr., E. G., and Rivest, R. L. (1980). Orthogonal Packings in Two Dimensions. *SIAM Journal on Computing*, 9(4):846–855.

[2] Gaticia, G., Reyes, P., Contreras-Bolton, C., Linfati, R., and Escobar, J. W. (2016). Un algoritmo para el Strip Packing Problem obtenido mediante la extracción de habilidades de expertos usando minería de datos. *Ingeniería, Investigación y Tecnología*, 17(2):179–190.

[3] Zhang, D., Che, Y., Ye, F., Si, Y.-W., and Leung, S. C. (2016). A hybrid algorithm based on variable neighborhood for the strip packing problem. *Journal of Combinatorial Optimization*, 32(2):513–530.

[4] Wei, L., Wang, Y., Cheng, H., and Huang, J. (2019). An open space based heuristic for the 2D strip packing problem with unloading constraints. *Applied Mathematical Modelling*, 70:67–81.

[5] Martin, M., Morabito, R., and Munari, P. (2020). A bottom-up packing approach for modeling the constrained two-dimensional guillotine placement problem. *Computers and Operations Research*, 115:104851.

[6] Gendreau, M., Iori, M., Laporte, G., and Martello, S. (2008). A Tabu Search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18.

[7] Christofides, N. and Whitlock, C. (1977). An Algorithm for Two-Dimensional Cutting Problems. *Operations Research*, 25(1):30–44.

[8] Burke, E. K., Kendall, G., and Whitwell, G. (2004). A New Placement Heuristic for the Orthogonal Stock-Cutting Problem. *Operations Research*, 52(4):655–671.

[9] Bengtsson, B. E. (1982). Packing rectangular pieces - a heuristic approach. *Computer Journal*, 25(3):353–357.

[10] Hopper, E. and Turton, B. C. (2001). A review of the application of meta-heuristic algorithms to 2D strip packing problems. *Artificial Intelligence Review*, 16(4):257–300.

[11] Hopper, E. and Turton, B. C. H. (2002). Problem Generators for Rectangular Packing Problems. Technical report.

[12] Beasley, J. E. (1985a). Algorithms for unconstrained two-dimensional guillotine cutting. *Journal of the Operational Research Society*, 36(4):297-306.

[13] Beasley, J. E. (1985b). An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. *Operations Research*, 33(1):49-64.

[14] Macqeen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability*, 1(66):281-297.

[15] Ketchen Jr., D. J. and Shook, C. L. (1996). The Application of Cluster Analysis In Strategic Managment Research: An Analysis And Critique. *Strategic Management Journal*, 17(6):441–458.