An analysis of the state of the art of algorithms applied to BPP

Joaquín Pérez Ortega[1], Hilda Castillo Zacatelco[2], and Rafael de la Rosa Flores[2]

[1] Centro Nacional de Invesatigación y Desarrollo Tecnológico, Departamento de Ciencias Computacionales, México,
jperez@cenidet.edu.mx
[2] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México,
hildacz@gmail.com, rafael@cs.buap.mx

Abstract. Before proposing a solution to the problem of bin packing it is important to know the state of the research. This paper presents a survey of the state of the art of those algorithms that according to the results reported in the literature, turn out to be the works that more instances have solved and therefore they turn out to be the most relevant. In this work we have classified them by the type of algorithm that they used as its base.

1 Introduction

There are a variety of real problems in areas such as health, industry, computing [6], among others that can be modeled as a bin packing problem (BPP). In particular, manufacturing industry, it is common the need to find ways to accommodate objects of different weights and that the space required is the minimum. Find the solution how to accommodate these objects allows to optimize production costs, time, saving material resources and other benefits. This motivated us to develop new techniques and algorithms, with different approaches to solve this problem.

The BPP is a combinatorial problem classified as NP-hard [19] and consists of placing a set of items within bins such that the number of bins is minimal. To date, there are different descriptions of the BPP, some of the most important are for example Coffman at al[8] defines the BPP using normalized values in the interval $(0, 1]$ and the bin size is 1. Although in most of the work both the capacity of the bin as the weight of the items are integers greater than zero. A formal description is carried out by Martello and Toth [27] raised the BPP as a problem linear programming. This approach consists in allocating items to a container such that the sum of the weights of items in each container does not exceed capacity C, and the number of containers is minimized. Currently there are algorithms that solve some special cases of the BPP, but there are many other cases that have not yet been satisfactorily resolved and are topics for research.

This article is a summary of a larger work that took place about the state of the art analysis of BPP that performs a classification of the algorithms that have solved the largest number of instances of the problem of BPP based on the type of algorithm that was used to resolve. In Coffman [8] makes an analysis of approximation algorithms known as offline and online and present a worst case analysis and average case of them and the date of publication of the paper were presented new algorithms that solve the BPP.

The paper is organized as follows: Section 2 presents a brief description of the theoretical framework of BPP. Section 3 presents an analysis of the state of the art of the problem and Section 4 we show the conclusions.

2 Conceptual theory for the BPP

To date have been developed exact algorithms and approximation algorithms and in recent times have focused on the use of hybrid algorithms. The following sections explain some of the most representative algorithms that solve the BPP

2.1 The computational complexity

One problem is tractable if there is a deterministic polynomial time algorithm to solve it, such problems belong to the class P [26]. Intractable problems are those that no polynomial time algorithm has been able to solve them. The class NP includes all the intractable problems [19].

A decision problem is NP-complete if it belongs to the class NP and it is possible to polynomially reduce any NP problem to this problem[5]. The Class NP problems that are intractable are called NP-hard.

A combinatorial optimization problem is one where each feasible solution there is an associated value and want to find a feasible solution with the best value (maximum or minimum) [6]. The BPP is a combinatorial optimization problem is considered as a problem NP-hard

2.2 Indexes of complexity and Classical heuristics

An index of complexity is a metric that measures the complexity of the problems [28]. For the BPP have generated a series of indicators that characterize instances of the problem and its behavior with different algorithms. [6] analyzed the following indexes: size, capacity occupied by an average item, the dispersion of size of the object among other. [1] proposes two additional indexes, which are taken up by the work done by [26] and adds one more. Within her master's thesis [35] makes an analysis of complexity indexes proposed to date and select a subset of them

In the state of the art are described several heuristics to obtain an approximate solution of the BPP, for example. Online heuristic as First Fit (FF), Best Fit (BF) and Next Fit (NF) [8] pack the items as they arrive. The FF heuristic takes an object from a list and open a bin, each object is assigned to a bin and removed from the list, if the next object to be placed in the bin exceeds the capacity of this, then this object is not placed in the bin, but opens a new bin and placed, the remaining objects are assigned to open bin. BF heuristics an object is assigned to the bin if this best fit, if there is no bin is then opened a new bin. The NF heuristic is similar to FF, but in this case, when a bin is opened, the objects are assigned one by one in the bin, when an object is no longer assigned to the bin then opens a new bin and continues with the process. There are other heuristics such as The first Fir Decreasing (FFD), Best Fit Decreasing (BFD) and Next Fit Decreasing (NFD), a good study of these heuristics can be found in [8]. In [13], [16], [22] and [37] other heuristics are proposed.

2.3 Lower bound

The lower or higher levels, as its name suggests, define the values that a variable can take, in this case the variable represents the minimum number of bins needed to store n items. In particular Martello and Toth [27] proposed three lower bounds:

L_1, provides the smallest number of bins that can accommodate items [34], this type of bound has a better behavior for instances where the weights of the items are sufficiently small with respect to the capacity of the bin. L_2, this type of bound is particularly useful when there are items with weights which exceed half the capacity of the bin. L_3 This limit is useful when there is both large weighted items, as items with small weights and is based on the procedure of Martello and Toth reduction [27], which is defined as follows: given two different feasible sets F_1 y F_2, if a partition F_2 into subsets $P_1, \ldots, P_t$ and a subset $\{j_1, \ldots, j_t\}$ of F_1 such that there $w_{j_h} \geq \sum_{k \in P_h} w_k$ for $h = 1, \ldots, t$, then F_1 dominate F_2.

Other bounds that are based in the Martello and Toth [27] bounds are those proposed by the work of Scholl [32], called BISON, he proposed the lower bounds: $L_4, L_5 y L_6$

2.4 Sets of test instances

According to the literature, sets of test instances that are used most often are: *Uniform* and *Triplets* proposed by [10], *Triplets* instances are considered difficult instances. *Data Set_1*, *Data Set_2* and *Data Set_3* proposed by [32], *Data Set_3* instances are considered difficult instances. *Was_1* and *Was_2* proposed by [34]. *Gau_1* proposed by [37]. Other sets of instances classified as hard and extremely difficult are *Extremely-ffd-hard* and *ffd-hard* proposed by [34], these instances were used in the work of [2] y [3]

3 Classification of works related to BPP

The use of pure exact algorithms to solve the BPP is impractical, so much of work dedicated to solving the BPP make use of heuristics and metaheuristics only. However, it has been observed that the use of exact methods combined with heuristics gives very good results [3].

The review state of the art has allowed to classify the work that solve the problem of bin packing in two groups, those that use deterministic algorithms and those that use non-deterministic algorithms. The classification of the works was performed taking into account the type of algorithm that essentially used the work.

The deterministic algorithms are those in which at each step of the algorithm determines the next step is unique for each input and the output is always the same. A nondeterministic algorithm is one that should decide at each step of the execution among several alternatives and examine them all before finding the solution.

In this paper will be understood as an exact algorithm to those algorithms that given the same input obtain the same result, they do not use heuristics and they have demonstration. [15]. It will be understood as heuristic to those empirical rules used to solve algorithms, they provide good results, but they have no demonstration. The combination of local search heuristics form the initial basis on which some algorithms developed metaheuristics. To be called metaheuristics the strategies of resolution of a general character of orientation to the search which no concrete all the details related to this, and that they can use heuristics. Randomized algorithm to be called that which under the same input the output can vary, not always solve the problem correctly and uses some random element (random numbers, random choices).

Among the deterministic algorithms are found the exact algorithms and the deterministic heuristic algorithms. Within the nodeterministic algorithms are the randomized heuristic algorithms and the nonrandomized heuristic algorithms. Among the nondeterministic algorithms are found the randomized heuristic algorithms and the nonrandomized algorithms.

In addition, there is a set of algorithms called metaheuristics, such algorithms can be use both randomized heuristics and deterministic heuristics, but at least one.

3.1 Heuristic deterministic algorithms

The best known and cited heuristics in bin packing papers are First Fit (FF), First Fit Decreasing (FFD) [19] and Best Fit Decreasing (BFD). They are all the most fundamental basis of the papers developed on this subject.

Garey, Graham and Ullman [17] in his article published en 1973 did an analysis of the FFD algorithm and they found that a decrease in the values of the weights of items can result in an increase in the number of bins required even an object can also cause an increase in the number of bins, so it was necessary to seek other alternatives.

Gupta and Ho [16] in 1999 presented a heuristic called minimun bin slack(MBS) which is focused on the bin. At each step a test is performed to find a set of items that match the capacity of the bin as much as possible. Each time a packing is ready, the items are assigned to a bin, placed on it and removed from the list.

The process is completed when the list becomes empty. Gupta and ho showed that MBS is superior in terms of quality solutions to FFD and BFD algorithms.

Fleszar and Hindi [13] proposed the following modification to MBS: before the search procedure is invoked, an item is selected and fixed permanently in a bin. This object will be the seed and it is the largest object, which leave the least waste into the bin to fill. In addition to the MBS' heuristic, Fleszar and Hindi [13] proposed other heuristics as MBS' relaxed, perturbation MBS', sampling MBS' and enhancements to the procedure of searching the neighborhood variable (VNS).

Byung-In Kim and Juyoung Wy [22] presented a paper in 2010 which they proposed four algorithms to improve the algorithms NFD, FFD and BFD. The algorithms are: FFD_L2F, BFD_L2F, NFD_L2F, and BBB_FFD_L2F.

Fleszar and Charalambous [11] in 2011 proposed new heuristics which they called "Bin oriented heuristics", these heuristics make use of the average weight of the items, in this form they to control its packaging and to avoid that many objects that have small weights are used from the beginning, which can result in poor solutions.

As we can see the heuristics presented by Garey are the main base of almost all current heuristics.

3.2 Exact deterministic algorithms

Martello and Toth [27] in 1990 published their book entitled "Knapsack Problems", which has served as the basis for subsequent works. For the problem of BPP, these researchers proposed the use of lower bounds (L_1, L_2 y L_3), in addition to the limits they proposed a reduction procedure called reduction procedure of Martello and Toth (MTRP) based on the criterion of Martello and Toth domination.

Schwerin and Wäscher [34] proposed in 1998 a new lower bound for the BPP, this heuristic was called L_{CS} and it is derived from standard optimization method for the cutting stock problem. It can be integrated to Martello and Toth method (MTP). The integration of the L_{CS} and MTP was called MTPCS.

Valerio de Carvalho [36] in 1999 formulated an exact solution algorithm for BPP using column generation and the branch and bound algorithm.

Sándor P. Fekete and Jörg Schepers [14] presented in 2001 a work in which they propose new classes of fast lower bounds for BPP. This proposal is based on dual feasible functions and can be interpreted as a generalization of L_2.

In 2010 Jarboui et. al [21] presented a new scheme for lower bound for the BPP in one dimension based on what are called destructive bounds.

In 2004 Cruz [6], under the direction of Dr. Joaquín Pérez Ortega of CENIDET, developed a work in where she did a selection of algorithms for BPP. To carry out this choice she was based on what is called indexes of complexity. Later in the year 2006 Alvarez [1] also addressed the issue of indexes of complexity and propose two more indexes. In 2007 Nieto [28] presented her masters thesis where she developed a high performance metaheuristic algorithm that integrates several approximate algorithms for solving the BPP. The basis of the algorithm is a genetic algorithm. Quiroz [35] on her masters thesis did a choice of indexes of complexity which really characterized an instance of the BPP. She also proposed other indexes.

Analyzing the work mentioned above we can see that most of the exact algorithms used branch and bound or column generation. It has also been increased the research in the development of new lower bounds and the characterization of BPP instances.

3.3 Metaheuristic algorithms

Falkenauer and Delchambre in 1992 [12] presented a grouping genetic algorithm that uses genetic operators other than used by the classical genetic algorithm and a new coding. Later, in 1996 Falkenauer [10] presented a hybrid algorithm for the BPP algoritm, which uses the algorithm of Falkenauer and Delchambre [12] called GGA and the criterion of dominance of Martello y Toth [27].

In 2004 J. Levine and F. Ducatelle [24] proposed a hybrid algorithm which uses ant colony optimization and local search and Bhatia and Basu [4] proposed a heuristic called better-fit and they implemented this heuristic in a genetic algorithm that uses a multi-chromosome representation. In 2006, Alok Singh and Ashok K. Gupta [31] proposed two heuristics for solving the BPP, these heuristics was called H-SGGA.

In 2007, Cruz Reyes et. al [7] presented a new hybrid intelligent system to solve bin packing. The methodology involves the fusion of soft computing using a genetic algorithm and Hard Computing using limits and deterministic strategies.

Kok-Hua Loh, Bruce Golden and Edward Wasil [25] proposed a new procedure that called WABP. This procedure implements the concept of weight annealing metaheuristic. In 2008 Stawowy [33] presented a simple non-hybrid evolutionary heuristic. This heuristic makes use of the FF algorithm to obtain the initial solution. The cost function used in this work is similar to that proposed by Delchambre and Falkenauer [12]. In the same year *Ü*lker et. al [9] proposed a genetic algorithm that uses Linear Linkage Encoding (LLE) as a representation scheme.

Pedro Gomez-Meneses and Marcus Randall [20] proposed a hybrid algorithm based on a concept called Extremal Optimization with an improvement in the local search for the BPP.

In 2010 Rohlfshagen [30] published an article that presents two genetic algorithms to solve the problem of bin packing and the problem of multiple knapsack. These algorithms are inspired by molecular genetics.

In the case of the use of metaheuristics the works have focused on the use of evolutionary algorithms such as genetic algorithms differing in the way that employs the operators and the way that objects and bins are represented.

3.4 Hybrid Algorithms

Scholl, Klein and Jurgens in 1997 [32] presented an exact algorithm called BISON. This algorithm uses the branch and bound algorithm, which uses new branching schemes with various cuttings and reduction procedures and heuristics. In her article published in 2001 [2], Alvim proposes an improved hybrid procedure based on the progressive increase in the number of bins used by a possible feasible solution.

4 Conclusions and future work

The majority of the works that solve bin packing not only focus on one-way, but most of them use several strategies. However, the classification performed here is based upon the essential algorithm used by the work. See 1.

Although the works that solve the problem of bin packing make use of different approaches, none of them have succeeded in resolving the whole set of instances of the benchmark tests.

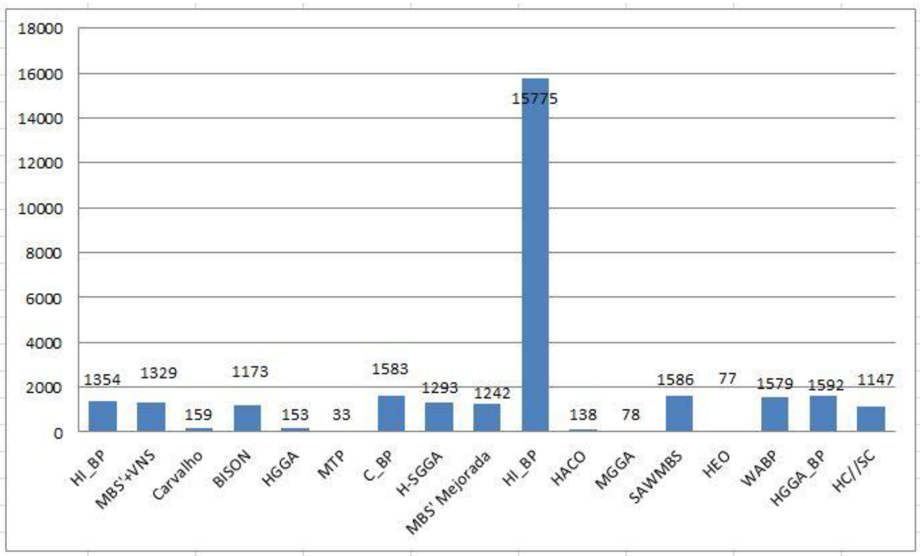

Fig. 1. Graph of total instances solved for each work.

In the case of works that make use of exact algorithms, these are limited to solving instances in which the number of items to be packaged is reduced and those who achieve solve large instances have huge solution times and many of them have been surpassed by heuristic algorithms. On the other hand the instances that have been solved by exact algorithms belong to the class of Uniform instances, Triplets, set_1, set_2 and set_3, but has not been reported to have solved instances of other classes that have been classified as "difficult" [29] .

In the case of works that make use of randomized heuristic algorithms and metaheuristics which includes randomized heuristics, have failed to resolve a large number of instances in a shorter time than exact and with a larger number of objects. The problem of such algorithms is that given the same input, the output may be different, so that to find the solution, the program must be execute multiple times.

Work	Deterministics algorithms				Nodeterministic algorithms
	Bounds	Exacts	Indexes	Heuristics	Heuristics
MTP (Martello, 1990)	■	■			
HGGA (Falkenauer, 1996)				■	■
BISON (Scholl, 1996)	■	■		■	■
Límites inferiores (Fekete, 1998)	■				
MTPCS (Schwerin, 1998)	■	■			
Column generation and branch and bound (Carvalho, 1999)		■			
MBS (Gupta, 1999)				■	
Nuevas heuristicas MBS (Fleszar, 2002)				■	
MGGA+BetterFit (Bhatia, 2004)				■	■
HACO (Levine, 2004)				■	■
HI_BP (Alvim, 2004)	■	■		■	■
C_BP (Singh, 2006)				■	■
HC//SC (Cruz, 2007)				■	■
HGGA_BP (Nieto, 2007)	■		■	■	■
WABP (Loh, 2008)				■	■
Evolutivo (Stawowy, 2008)				■	■
LLE (Ülker, 2008)				■	■
HGGA_BP Mejorado (Quiroz, 2009)	■		■	■	■
HEO (Gómez, 2009)				■	■
Límites inferiores (Jarboui, 2010)	■				
Caracterización de instancias (Pérez, 2010)			■		
L2F (Kim, 2010)				■	
Pert-SAWMBS (Fleszar, 2011)				■	■

Table 1. Classification of works.

However, the works that make use of several deterministic algorithms and nondeterministic algorithms have given the best results, as demonstrated by Alvim at her work, although this can not be said that the bin packing problem has been resolved, as many instances test are still unresolved. Table 1 presents a classification of the studies reviewed in the state of the art, the classification is according to the classification algorithms described above.

The Figure 1 shows the graph of the total number of instances has solved each job.

Analyzing the data in Tables 2, 3 and 4 we can see that the researchers did not match exactly in terms of the sets of instances used to measure the efficiency of their algorithms.

In Tables 2, 3 and 4 show the number of instances per dataset that some jobs have been resolved.

Instance	Number of instances	HI_BP (Alvim, 2002)	Perturbation MBS' + VNS (Fleszar, 2002)	Column generation and branch and bound (Carvalho, 1999)	BISON (Scholl, 1996)	HGGA (Falkenauer, 1996)	MTP (Martello, 1990)
U120	20	20	20	20		18	18
U250	20	20	19	20		20	9
U500	20	20	20	19		20	
U1000	20	20	20	20		20	
U2000	20						
U4000	20						
U8000	20						
T60	20	20	20	20		18	6
T120	20	20	20	20		20	
T249	20	20	20	20		20	
T501	20	20	20	20		20	
Set_1	720	704	694		697		
Set_2	480		474				
Set_3	10	10	2		3		
Extremely-ffd-hard	1900, 3600						
Ffd-hard	1600, 2200, 3700, 1500						
Was_1	100						
Was_2	100						
Gau_1	17						

Table 2. Datasets solved by different algorithms.

Based on the literature, the work of Alvim [3] reported to have solved the largest number of instances of the benchmark, for a total of 15775 of 16138 instances, 363 instances remaining unresolved. Later in the work of Quiroz was able to solve 6 instances of the 363 that did not solve Alvim. However, this algorithm was not capable of solve all instances that the algorithm had already resolved Alvim. To date, in the subsequent improvements have been achieved in the execution times of the instances already solved by the algorithms already mentioned. In spite of the above have not yet been able to resolve 357 instances of the benchmark.

On the other hand, analyzing the graph of the Figure 1, we see that the best works to date, regarding the number of instances solved, are the work of Alvim [3], the work of Fleszar [11] and the work of Quiroz [35]. One important point to be emphasized in this figure, is that it can be seen that the most relevants are those that are hybrids and / or using metaheuristics that had not been used thus far to solve the problem of bin packing.

In recent years, researchers have worked on new heuristics and metaheuristics , and in the study of lower bounds for bin packing.

Instance	Number of instances	C_BP (Singh, 2006)	H-SGGA (Singh, 2006)	Perturbation MBS' Mejorada (Singh, 2006)	HI_BP (Alvim, 2004)	HACO (Levine, 2004)	MGGA + BetterFit (Bhatia, 2004)
U120	20	20	20	18	20	20	20
U250	20	19	18	16	20	18	18
U500	20	20	20	19	20	20	20
U1000	20	20	20	19	20	20	20
U2000	20					20	
U4000	20					20	
U8000	20					20	
T60	20	20	6	20	20		
T120	20	20	0	20	20		
T249	20	20	1	20	20		
T501	20	20	2	20	20		
Set_1	720	719	717	662	720		
Set_2	480	480	479	428	480		
Set_3	10	10	10		10		
Extremely-ffd-hard	1900, 3600				1900, 3567		
Ffd-hard	1600, 2200, 3700, 1500				1600, 2183, 3568, 1375		
Was_1	100	100			100		
Was_2	100	100			100		
Gau_1	17	15			12		

Table 3. Datasets solved by different algorithms (continuation).

Instance	Number of instances	Pert-SAWMBS (Fleszar, 2011)	HEO (Gómez, 2009)	WABP (Loh, 2008)	HGGA_BP Mejorado (Quiroz, 2009)	HC//SC (Cruz, 2007)
U120	20	20	20	20		
U250	20	20	17	20		
U500	20	20	20	20		
U1000	20	20	20	20	79 de 80	
U2000		20				
U4000		20				
U8000		20				
T60	20	20		20	20	
T120	20	20		20	20	
T249	20	20		20	20	
T501	20	20		20	20	
Set_1	720	720		720	718	661
Set_2	480	480		480	480	442
Set_3	10	10		10	9	
Extremely-ffd-hard	1900, 3600					
Ffd-hard	1600, 2200, 3700, 1500					
Was_1	100	100		100		
Was_2	100	100		100		
Gau_1	17	16		9		
Hard28	28				8	
NIRUP	53				3	

Table 4. Datasets solved by different algorithms (continuation).

References

1. Víctor Manuel Alvarez Hernández. *Modelo para representar la complejidad del problema y el desempeño de algoritmos.* Instituto Tecnológico de Ciudad Madero, 1 edition, 2006.

2. Adriana C.F. Alvim, Fred Glover, Celso C. Ribeiro, and Dario J. Aloise. A hybrid improvement heuristic for the bin packing problem. *Metaheuristics International Conference*, 8:63–68, 2001.
3. Adriana C.F. Alvim, Fred Glover, Celso C. Ribeiro, and Dario J. Aloise. A hybrid improvement heuristic for the bin packing problem. *Journal of Heuristics*, 10:205–229, 2004.
4. A. K. Bhatia and S. K. Basu. Packing bins using multi-chromosomal genetic representation and better-fit heuristic. *Neural Information Processing, 11th International Conference, ICONIP 2004, Calcutta, India, November 22-25, 2004, Proceedings*, 3316:181–186, 2004.
5. G. Brassard and P. Bratley. *Fundamentos de algoritmia.* Pearson, Prentice Hall, 1 edition, 2008.
6. Laura Cruz Reyes. *Clasificación de algoritmos heurísticos para la solución de problemas de bin packing.* Instituto Tecnológico de Ciudad Madero, 1 edition, 2004.
7. Laura Cruz Reyes, Diana Maritza Nieto Yañez, Pedro Tomás Solis, and Guadalupe Castilla Valdez. Solving bin packing with a hybridization of hard computing and soft computing. *Innovations in Hybrid Intelligent Systems*, 2007.
8. E. G. Coffman Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. *Approximation Algorithms for NP-Hard Problems, D. Hochbaum*, 1997.
9. Özgür Ülker, Emin Erkan Korkmaz, and Ender Özcan. A grouping genetic algorithm using linear linkage encoding for bin packing. *Springer-Verlag Berlin*, 2008.
10. E. Falkenauer. A hybrid grouping genetic algorithm for the bin packing. *Journal of Heuristics*, 2:5–30, 1996.
11. Krzysztof Fleszar and Christoforos Charalambous. Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operational Research*, 210:176–184, 2011.
12. E. Falkenauer and A. Delchambre. A genetic algorithm for bin packing and line balancing. *Proceedings of the 1992 IEEE Internacional*, 2:1186–1192, 1992.
13. Krzysztof Fleszar and Khalil S. Hindi. New heuristics for one - dimensional bin packing. *Computers & Operations Research*, 29:821 – 839, 2002.
14. Sándor P. Fekete and Jörg Schepers. New classes of fast lower bounds for bin packing problems. *Math. Program.*, 91:11–31, 2001.
15. Carlos P. Gracia Calandín. *Métodos y algoritmos para resolver problemas de corte unidimensional en entornos realistas. Aplicación a una empresa del sector siderúrgico.* Universidad Politécnica de Valencia, 1 edition, 2008.
16. Jatinder N. D. Gupta and Johnny C. Ho. A new heuristic algorithm for the one-dimensional binpacking problem. *Production Planning & Control*, 10:598–603, 1999.
17. M. R. Garey, R. L. Graham, and J. D. Ullman. An analysis of some packing algorithms. *In Combinatorial Algorithms*, 1973.
18. M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness.* Bell Laboratories, 1 edition, 1979.
19. M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness.* Bell Laboratories, 1 edition, 1979.
20. Pedro Gómez-Meneses and Marcus Randall. A hybrid extremal optimisation approach for the bin packing problem. *ACAL, 2009, Springer-Verlag Berlin*, 2009.
21. Bassem Jarboui, Saber Ibrahim, and Rebai Abdelwahen. A new destructive scheme for the bin packing problem. *Ann. Operations Research*, 179:187–202, 2010.

22. Byung-In Kim and Juyoung Wy. Last two fit augmentation to the well-known construction heuristics for one-dimensional bin-packing problem: an empirical study. *Int. J. Adv. Manuf. Technol.*, 50:1145–1152, 2010.
23. Vanesa Landero Nájera. *Metodología para la explicación causal del comportamiento y desempeño de algoritmos heurísticos aplicdos al problema de bin packing*. Cenidet, 1 edition, 2008.
24. J. Levine and F. Ducatelle. Ant colony optimization and local search for bin packing and cutting stock problems. *The Journal of the Operational Research Society*, 55:705–716, 2004.
25. Kok-Hua Loh, Bruce Golden, and Edward Wasil. Solving the one-dimensional bin packing problem with a weight annealing heuristic. *Computers & Operations Research*, 35:2283–2291, 2008.
26. Vanesa Landero Nájera. *Metodología para la explicación causal del comportamiento y desempeño de algoritmos heurísticos aplicdos al problema de bin packing*. Cenidet, 1 edition, 2008.
27. S. Martello and P. Toth. *Knapsack Problems. Algorithms and Computer.* John Wiley & Sons, 1990.
28. Diana Maritza Nieto Yañez. *Hibridación de algoritmos metaheurísticos para problemas de bin packing*. Instituto Tecnolgico de Cd. Madero, 1 edition, 2007.
29. Joaquín Pérez, David Romero, Laura Cruz, Adriana Mexicano, Rodolfo Pazos, Gerardo Reyes, and René Santaolaya. Characterization of bin packing problem hard instances. *ALIO/INFORMS International Meeting*, 2010.
30. Philipp Rohlfshagen and John A. Bullinaria. Nature inspired genetic algorithms for hard packing. *Annals of Operation Research*, 179:393–419, 2010.
31. Alok Singh and Ashok K. Gupta. Two heuristics for the one - dimensional bin packing problem. *Operations Research*, 8:101–111, 2006.
32. Armin Scholl, Robert Klein, Bruce Golden, and Christian Jürgens. Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research*, 24:627–645, 1997.
33. Adam Stawowy. Evolutionary based heuristic for bin packing problem. *Computers & Industrial Engineering*, 55:465–474, 2008.
34. Petra Schwerin and Gerhard Wäscher. A new lower bound for the bin packing problem and its integration into mtp. *Springer-Verlag*, 1998.
35. Marcela Quiroz Castellanos. *Caracterización de factores de desempeño de algoritmos de solución de BPP*. Instituto Tecnológico de Ciudad Madero, 1 edition, 2009.
36. J. M. Valerio de Carvalho. Exact solution of bin packing problems using column generation and branch and bound. *Annals of Operation Research*, 86:629–659, 1999.
37. G. Wäscher and T. Gau. Heuristics for the integer one-dimensional cutting stock problem: A computational study. *OR Spectrum*, 18:131–144, 1996.