Comparative Study of Heuristics for the One-Dimensional Bin Packing Problem

Chapter · June 2023

DOI: 10.1007/978-3-031-28999-6_19

7 authors, including:

Jessica Elena González San Martín
Instituto Tecnológico de Ciudad Madero, (ITCM)

Laura Cruz-Reyes
Instituto Tecnológico de Ciudad Madero, (ITCM)

Claudia Guadalupe Gómez Santillán
Instituto Tecnológico de Ciudad Madero, (ITCM)

Héctor Joaquín Fraire-Huacuja
Instituto Tecnológico de Ciudad Madero, (ITCM)

Comparative study of heuristics for the One-dimensional Bin Packing Problem

Jessica González-San-Martín[1], Laura Cruz-Reyes[1], Claudia Gómez-Santillán[1], Héctor Fraire[1], Nelson Rangel-Valdez[2], Bernabé Dorronsoro[3] and Marcela Quiroz-Castellanos[4]

[1]Tecnológico Nacional de México / Instituto Tecnológico de Ciudad Madero, Cd. Madero, Tamaulipas, México.

[2]CONACyT Research Fellow at Graduate Program Division, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Cd. Madero, Tamaulipas, México.

[3]Computer Science Engineering Department University of Cadiz, Puerto Real, Cadiz, Spain.

[4]Artificial Intelligence Research Center Universidad Veracruzana, Xalapa-Enriquez, Veracruz, México.

Abstract The Bin Packing problem (BPP) is a classic optimization problem that is known for its applicability and complexity, which belongs to a special class of problems called NP-hard, in which, given a set of items of variable size, we search to accommodate them inside fixed size containers, seeking to optimize the number of containers to be used, that is, using the least number of containers to place the largest number of items possible. The BPP has been preserved as a current study problem due to the various applications that it offers mainly in the industry; therefore, in the recent state-of-the-art, there are different algorithms, mainly heuristics and metaheuristics for solving the problem. In this paper, we present an empirical comparison of the algorithms that have been used to solve one of the variants of BPP; the One-dimensional Bin Packing Problem (1D-BPP), and the works that have reported the best results in the state-of-the-art, as well as those found in literature from today to the last two decades. This survey aims to identify which components and techniques were used for each of the algorithms and which of these contribute more to their performance. Twenty-one algorithms were selected from the specialized literature that have the aforementioned characteristics, their results were analyzed with different instances and the methods they use were added, for example, neighborhood and local searches, evolutionary and genetic algorithms, among others. The main objective is that this study can help both researchers and professionals interested in using specific components or techniques that help improve the behavior of an algorithm to solve a problem, in this case, 1D-BPP, since we hope that our conclusions can provide some ideas about the advantages or limitations of each of the methods studied here.

Keywords Bin packing · Techniques · heuristics and metaheuristics

1. Introduction

The One-dimensional Bin Packing Problem (1D-BPP) is a challenging NP-Hard combinatorial optimization problem [1] where each object has a unique weight, and the objective is to pack the greatest number of objects in the least number of possible bins. Martello and Toth [2] formally defined the 1D-BPP as follows:

$$minimize \sum_{k=1}^{K} y_k \tag{1}$$

$subject\ to$

$$\sum_{k=1}^{K} x_{ik} = 1, \qquad \forall i \in I \tag{2}$$

$$\sum_{i=1}^{n} w_i x_{ik} \leq w_{y_k}, \qquad \forall k \in K \tag{3}$$

$$y_k \in \{0,1\}, \qquad \forall k \in K \tag{4}$$

$$x_{ik} \in \{0,1\}, \qquad \forall i \in I, \forall k \in K \tag{5}$$

where n is the number of items, w_i is the weight of the item i, W is the bin capacity, $I = \{1, \dots, n\}$ is the set of items, $K = \{1, \dots, K\}$ is the set of bins, the variables x_{ik} and y_{ik} are defined as:

$$x_{ik} = \begin{cases} 1 & \text{if item } i \text{ is assigned to bin } k, \\ 0 & \text{otherwise} \end{cases} \qquad y_k = \begin{cases} 1 & \text{if bin } k \text{ is used,} \\ 0 & \text{otherwise} \end{cases}$$

This problem has been preserved as a current case study due to its complexity and the various applications it offers. In recent decades, different works related to 1D-BPP have been published. In the state-of-the-art we can find different types of algorithms, mainly heuristics and metaheuristics that focus on solving One-dimensional Bin Packing, however, some of the exact algorithms proposed in the literature are also contemplated.

In this study, we present an empirical comparison of the algorithms and the strategies that have been used to solve this problem, especially the algorithms that report the best results. Our paper contemplates the most recent works up to those presented twenty years ago, to show the evolution of the strategies and approaches used in this line of research.

2. One-dimensional Bin Packing techniques

This section review 1D-BPP techniques proposed in the past twenty years and report their strategies and strengths. The selection method used was based on the results reported by its authors and the number of quotes in the articles. We present twenty-one works among which we can find heuristics, metaheuristics, and some exact algorithms.

2.1 Heuristic techniques

Over time, different heuristics have been proposed to solve the 1D-BPP. Most of these works use improvements or variants of classical heuristics that are used to address this problem. As is known, heuristics are generally scalable and very fast in generating solutions, however, optimality is not guaranteed [3].

Fleszar and Hindi [4] developed a high-quality algorithm (Perturbation-MBS') that incorporates a modified version of the MBS heuristic from Gupta and Ho [5], a procedure that is based on a variable neighborhood search and lower bounds, to successively build a new solution by perturbing the current solution.

Alvim et al. [6] presented a hybrid improvement heuristic (HI_BP) in which the dual strategy used by Scholl et al. [7] is reintroduced, in addition to the search space reduction techniques of Martello and Toth [2] and the use of lower bounding strategies. This procedure uses a load redistribution method that is based on dominance, differencing, and unbalancing and the inclusión of an improvement process that utilizes a tabú search.

Singh and Gupta [8] proposed a combined approach (C_BP) that uses two heuristics: a hybrid steady-state grouping genetic algorithm and an improved version of the Perturbation-MBS heuristic from Fleszar and Hindi [4].

Loh et al. [9] developed a weight annealing procedure (WA), making use of the concept of weight annealing to expand and speed up the neighborhood search by creating distortions in different parts of the search space. The proposed algorithm is simple and easy to implement, and the authors reported a high-quality performance.

Fleszar and Charalambous [10] proposed a modification to the Perturbation-MBS method [4] that uses a new sufficient average weight (SAW) principle to control the average weight of the items that are packed in each bin (Perturbation-SAW-MBS).

Cruz-Reyes et al. [11] proposed a hybrid genetic clustering algorithm called HGGA-BP, which is inspired by the Falkenauer group representation scheme that applies evolutionary operators at the container level. This algorithm includes efficient heuristics to generate the initial population and perform group mutation and crossover; as well as hybrid strategies for the accommodation of objects that were left free when applying the group operators.

Sim and Hart [12] presented an island model that uses a form of cooperative co-evolution to generate a set of deterministic heuristics through single-node genetic programming, which interacts cooperatively to collectively minimize the number of containers used.

Buljubašić and Vasquez [13] offered a consistent neighborhood search approach for 1D-BPP. They used different techniques such as problem reduction through the first fit heuristic to obtain a partial solution, Taboo Search to add or remove elements from the containers, and the hill climbing/descent procedure to minimize the given objective function.

Table 1 Heuristic techniques for 1D-BPP

Author	Algorithm	Techniques	Year	Quotes
Fleszar and Hindi [4]	Perturbation-MBS	Variable neighborhood search and lower bounds.	2002	257
Alvim et al. [6]	HI_BP	Reduction techniques, dual and lower bounding strategies.	2004	192
Singh and Gupta [8]	C_BP	Hybrid steady-state grouping genetic algorithm and an improved version of the Perturbation-MBS heuristic [4].	2007	51
Loh et al. [9]	WABP	Weight annealing procedure and neighborhood search.	2008	95
Fleszar and Charalambous [10]	Perturbation-SAW-MBS’	Modification of the MBS method [4] that uses a new sufficient average weight (SAW).	2011	72
Cruz-Reyes et al. [11]	HGGA-BP	Genetic algorithm, different simple heuristics, and evolutionary operators at the container level.	2012	2
Sim and Hart [12]	GSMCH_BP	Single Node Genetic Programming, Hyper heuristics, and island models	2013	31
Buljubašić and Vasquez [13]	CNS_BP	Reduction techniques, a variant of First Fit heuristic and local searches: Taboo search and Hill Climbing Descent	2016	21

2.2 Metaheuristic techniques

Metaheuristic algorithms have become dominant for solving challenging optimization problems in various fields because they are widely used tools for finding (near) optimal solutions to solve large problem instances in reasonable execution times. Metaheuristic algorithms can be divided into evolutionary algorithms (EA) like the genetic algorithm (GA) or swarm intelligence algorithms like ant colony optimization (ACO), particle swarm optimization (PSO), and whale optimization, among others. Bhatia and Basu [14] introduced a multi-chromosomal grouping genetic algorithm (MGGA) with grouping genetic operators and a rearrangement procedure based on the better-fit heuristic.

Levine and Ducatelle [15] proposed a hybrid procedure that implements the ant colony optimization metaheuristic (HACO-BP), which includes a strategy for a local search that relies on the dominance criterion from Martello and Toth [2].

Layeb and Boussalia [16] presented an approach based on the quantum-inspired cuckoo search algorithm and define an appropriate quantum representation based on qubit representation to represent bin packing solutions. Also, they proposed a new hybrid quantum measure operation that uses the first fit heuristic to pack no filled objects by the standard measure operation.

Layeb and Chenche [17] presented an approach based on the GRASP procedure. They make use of two phases; the first is based on a new random heuristic based on the hybridization between the First Fit and Best Fit heuristics. while in the second phase the Taboo search algorithm is used to improve the solutions found in phase one.

Dokeroglu and Cosar [18] proposed a set of robust and scalable hybrid parallel algorithms that take advantage of parallel computation techniques, evolutionary grouping genetic metaheuristics, and bin-oriented heuristics to obtain solutions for large-scale one-dimensional BPP instances.

Bayraktar et al. [19] integrated an Artificial Bee Colony (ABC) algorithm and equipped it with a memory component that combines two main search methods for diversification and intensification: neighborhood search and random search. Memory integration is a component of a well-known metaheuristic, called Tabu Search, which steps up to neighborhood search with some awareness, where short-term and long-term memory models are recruited.

Quiroz-Castellanos et al. [20] propound the GGA-CGT algorithm that focuses on promoting the transmission of the best genes on the chromosomes while maintaining the balance between selection pressure and population diversity. The transmission of the genes is generated through a set of genetic clustering operators, while the evolution is balanced by a reproduction technique that controls the exploration of the search space and prevents premature convergence of the algorithm.

Kucukyilmaz and Kiziloz [21] proposed a scalable Island-parallel grouping genetic algorithm (IPGGA), where on a given run, the population is partitioned into semi-isolated subpopulations. With this parallelization approach, they manage to explore and exploit the search space more efficiently.

El-Ashmawi and Abd Elminaam [22] presented a modified version of the optimization algorithm (SSA) squirrel search algorithm. The proposed algorithm is based on an improved best-fit heuristic to generate a feasible initial packing of elements into containers. During solution improvement, operator strategy can take place to obtain an optimized solution for the BPP.

Borgulya [23] proposed a hybrid evolutionary algorithm where an individual is a feasible solution and contains the description of the containers. The algorithm works without recombination; uses two new mutation operators and improves the quality of solutions with local search procedures. The work of the mutation operators is based on a frequency matrix of relative pairs, and based on this matrix, the frequency of each pair of elements is known, that is, how often they are included in the same container in the best solutions. The frequency matrix helps to pack elements into subsets of elements; these subsets are the containers in the problem.

Hartono et al. [24] present a hybrid classical-quantum approach called H-BPP. The algorithm consists of two modules, each one designed to be executed in different computational ecosystems. First, a quantum subroutine searches for a set of feasible container configurations of the problem at hand. Second, a classical computation subroutine builds complete solutions to the problem from the subsets given by the quantum subroutine.

Table 2 Metaheuristic techniques for 1D-BPP

Author	Algorithm	Techniques	Year	Quotes
Bhatia and Basu [14]	MGGA	Genetic algorithm with grouping genetic operators and a rearrangement procedure based on the better-fit heuristic	2004	34
Levine and Ducatelle [15]	HACO-BP	Ant colony optimization metaheuristic and a strategy for a local search that relies on the dominance criterion from [2]	2004	289
Layeb and Boussalia [16]	QICSABP	Quantum-inspired cuckoo search algorithm containing a Gubit representation for the search space and quantum operators.	2012	72
Layeb and Chenche [17]	GRASP-BP	GRASP procedure and hybridization between the First Fit and Best Fit heuristics and Taboo search algorithm.	2012	19
Dokeroglu and Cosar [18]	Parallel Exon-MBS-BFD	Hybrid parallel algorithms, evolutionary grouping genetic metaheuristics, and bin-oriented heuristics.	2014	62
Bayraktar et al. [19]	MEABC	Integrated an Artificial Bee Colony (ABC) algorithm equipped it with a memory component that combines a neighborhood search (Taboo) and random search.	2014	6
Quiroz-Castellanos et	GGA-CGT	A genetic algorithm that focuses on promoting the transmission of the best genes	2015	103

al. [20]		on the chromosomes.		
Kucukyilmaz and Kiziloz [21]	IPGGA	Grouping genetic algorithm with parallel island model.	2019	42
El-Ashmawi and Abd Elminaam [22]	MSBPP	A squirrel search algorithm based on an improved best-fit heuristic.	2019	42
Borgulya [23]	HEA	A hybrid evolutionary algorithm that works without recombination. It uses two mutation operators.	2021	6
Hartono et al. [24]	H-BPP	A hybrid classical-quantum approach with a quantum subroutine searches for a set of feasible container configurations and a classical computation subroutine builds complete solutions.	2022	0

2.3 Exact techniques

The exact methods, as their name indicates, obtain the optimal solution to a problem, however, in instances that are very difficult to solve, the execution times of this type of algorithm are usually exponential. Nevertheless, in the state-of-the-art, few exact methods are used to address 1D-BPP.

Brandao and Pedroso [25] presented an exact method to solve bin packing and cutting stock problems that include multiple constraint variants. Their proposal is based on a laterally constrained arc-flow formulation equivalent to Gilmore and Gomory, including a graph compression algorithm that builds a graph, where paths from the source to the target node represent every valid packing pattern.

Wei et al. [26] proposed a branch-and-price-and-cut algorithm, this is another exact method based on the classical set-partitioning model for the One-dimensional bin packing problems and the subset row inequalities. They implemented an ad hoc label-setting algorithm as well as dominance and fathoming rules used to speed up its computation.

Table 3 Exact techniques for 1D-BPP

Author	Algorithm	Techniques	Year	Quotes
Brandao and Pedroso [25]	General Arc-Flow Formulation	Laterally constrained arc-flow formulation, including a graph compression algorithm that builds a graph.	2016	102
Wei et al. [26]	EXM	Branch-and-Price-and-Cut Algorithm formulated with Set-Partitioning and SR Inequalities	2020	31

3. Relevant optimization methods

We present a brief description of the three best works of the twenty-one reviewed with respect to the results reported by each author.

3.1 Grouping Genetic Algorithm with Controlled Gene Transmission

The Grouping Genetic Algorithm with Controlled Gene Transmission algorithm (GGA-CGT) developed by Quiroz-Castellanos et al. [20], is an intelligent packing metaheuristic that simplifies and improves the packing of objects, makes use of a rearrangement procedure that allows the exploration and exploitation of the search space and a set of genetic clustering operators that promote the transmission of the best genes on the chromosomes without losing the balance between selection pressure and population diversity. The transmission of the best genes (the fullest containers) is carried out through a set of genetic clustering operators, while the evolutionary process is balanced using a reproduction technique that controls the exploration of the search space and avoids the premature convergence of the algorithm.

Algorithm 1: GGA-CGT

Generate the initial population P with FF-$\tilde{n}$
while *generation* < *max_gen* y size(*best_solution*) > L_2
    Select n_c individuals to cross by Controlled_Selection
    Apply Gen_Level_Crossover_FFD to n_c selected individuals
    Apply Controlled_Replacement to introduce offspring
    Select n_m individuals and clone elite solutions by Controlled_Selection
    Apply Adaptative_Mutation_RP to the best n_m individuals
    Apply Controlled_Replacement to introduce clones
    Update *best_solution*
end while

3.2 Consistent neighborhood search for bin packing problem

The Consistent Neighborhood Search for one-dimensional Bin Packing (CNS_BP) algorithm is a heuristic proposed by Buljubašić and Vasquez [13] that consists of performing a local search to derive a feasible solution with a given number of bins, m, starting from $m = UB - 1$, where UB is an upper bound obtained by using a variant of the classic First Fit heuristic. It makes use of a reduction to simplify the problem and then applies two local searches: Taboo Search and Hill Climbing Descent. This algorithm shows successful results in its experimentations since it obtains equal or better solutions than the other algorithms in the state of the art, including the Hard28 instances that are considered one of the most difficult within BPP.

Algorithm 2: CNS_BP

Remove item pairs (i,j) such that $w_i + w_j = C$
Compute lower bound LB
Randomly shuffle the set of items
$S \leftarrow$ complete solution obtained by First Fit
$m \leftarrow$ number of bins in S
while $m > LB$ **and** time limit not exceded **do**
    $m \leftarrow m - 1$
    Build partial solution P with m-2 bins ▹ Delete 3 bins from S
    $S' \leftarrow CNS(P)$ ▹ Try to find complete solution with m bins
    If solution S' not complete **then** TERMINATE
    $S \leftarrow S'$
end while
return S

3.3 General Arc-Flow Formulation with Graph Compression

This work was proposed by Brandao and Pedroso [25], which is an exact method based on an arc flow formulation with lateral constraints. It is used to solve Bin Packing and Cutting Stock problems. It includes a graph compression algorithm that reduces the size of the underlying graph substantially without weakening the model. The formulation used is equivalent to that of Gilmore and Gomory [27]. However, instead of using column generation in an iterative process, the method builds a graph, where the paths from source to destination node represent each valid packing pattern.

4. Comparison of results

In the literature, different instances are used to solve the one-dimensional Bin Packing problem. These instances can be classified into easy and difficult [25]. Table 4 presents the instances that were used to evaluate the performance of the three most relevant algorithms of this study and Table 5 presents the comparative results.

Table 4 Set of test instances used by the most relevant algorithms in the literature.

Author	Class name	Number of instances
Falkenauer [1]	Uniform	80
	Triplets	80
Scholl et al. [7]	Data set 1	720
	Data set 2	480
	Data set 3	10
Schwerin & Wäscher [28,29], Gau [30]	Was 1	100
	Was 2	100
	Gau 1	17
Schoenfield [31]	Hard28	28

For the comparison of results, tests were carried out on the most relevant algorithms of those reviewed in this work; GGA-CGT, General Arc-Flow Formulation, and CNS_BP. For the first two algorithms, the experimentation was carried out in a Cluster with operating system CentOS version 6.7, GCC version 5.1.0, and Intel Xeon E5-2650 2.30GHz processor, while CNS_BP was executed under a virtual machine with operating system. system Ubuntu 14.04.4, GCC version 4.8.4, and Intel Celeron 2.16GHz processor, the latter was run in a virtual machine because it needed different conditions than the cluster.

Table 5 Comparison of results among the three most relevant works of the state of the art.

Class	**Inst.**	**CNS_BP**		**GGA-CGT**		**General Arc-Flow F.**	
		Opt.	T(s)	Opt.	T(s)	Opt.	T(s)
Uniform	80	80	0.07	80	0.23	80	0.16
Triplets	80	80	0.02	80	0.41	80	0.17
Data Set 1	720	720	0.07	720	0.35	720	3.33
Data Set 2	480	480	0.03	480	0.12	480	3.33
Data Set 3	10	10	0.00	10	1.99	10	3.33

Was 1	100	100	0.00	100	0.00	100	0.10
Was 2	100	100	0.00	100	1.07	100	0.10
Gau 1	17	17	2.68	16	0.27	17	9.58
Hard28	28	25	7.21	16	2.40	28	0.82
Total	1615	1612	10.08	1602	6.84	1615	20.92

We can see that the exact algorithm manages to solve the set of test instances optimally, however, as expected, it takes longer than the heuristic and metaheuristic algorithms. On the other hand, the CNS_BP heuristic manages to solve 99.81% of the instances in almost half the time that the exact algorithm takes.

5. Conclusion

This work has reviewed the algorithms developed in the last twenty years that have been used to solve the 1D-BPP. We find several works that make use of different classical heuristics as well as nature-inspired and population-based metaheuristics. The Metaheuristics approaches are widely used optimization tools for finding near-optimal solutions to large graph-sized problem instances of the BPP in reasonable execution times. These algorithms continue to be promising tools for numerous NP-Hard optimization problems where the exact solution methods tend to fail because of the exponential search spaces. However, some disadvantages of the metaheuristic algorithms include the issue of premature convergence, which can lead the algorithms into getting trapped in local optimum, and the aspect of parameter fine-tuning, as some of the algorithms would require having to set the control parameter to meet a certain specified threshold. It would be interesting to perform hybridization of metaheuristics with machine learning strategies, this would help to resolve certain limitations such as parameter tuning. It would also be an area of opportunity since, to our knowledge, no approach in the literature uses computational learning to solve the bin packing problem. We hope that with the experimental results provided, the best strategies can be taken to obtain more promising results if these are combined or implemented with some variation or improvement.

5. References

[1] Falkenauer Emanuel. 1996. A Hybrid Grouping Genetic Algorithm for Bin Packing. Journal of Heuristics 5–30.

[2] Martello, S., and Toth, P. 1990. Lower bounds and reduction procedures for the bin packing problem. Discrete Applied Mathematics 28 (1): 59–70.
[3] Ahmad, I., AlFailakawi, M. G., AlMutawa, A., & Alsalman, L. (2021). Container scheduling techniques: A survey and assessment. Journal of King Saud University-Computer and Information Sciences.
[4] Fleszar, K., & Hindi, K. S. (2002). New heuristics for one-dimensional bin-packing. Computers & operations research, 29(7), 821-839.
[5] Gupta, J. N., & Ho, J. C. (1999). A new heuristic algorithm for the one-dimensional bin-packing problem. Production planning & control, 10(6), 598-603.
[6] Alvim, A. C., Ribeiro, C. C., Glover, F., & Aloise, D. J. (2004). A hybrid improvement heuristic for the one-dimensional bin packing problem. Journal of Heuristics, 10(2), 205-229.
[7] Scholl, A., Klein, R., Jurgens, C., 1997. Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. Computers & Operations Research 24 (7), 627 _ 645
[8] Singh, A., & Gupta, A. K. (2007). Two heuristics for the one-dimensional bin-packing problem. OR Spectrum, 29(4), 765-781.
[9] Loh, K. H., Golden, B., & Wasil, E. (2008). Solving the one-dimensional bin packing problem with a weight annealing heuristic. Computers & Operations Research, 35(7), 2283-2291.
[10] Fleszar, K., & Charalambous, C. (2011). Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. European Journal of Operational Research, 210(2), 176-184.
[11] Cruz-Reyes, L., Quiroz C, M., F Alvim, A. C., Fraire Huacuja, H. J., Gómez S, C., & Torres-Jiménez, J. (2012). Efficient Hybrid Grouping Heuristics for the Bin Packing Problem. Computación y sistemas, 16(3), 349-360.
[12] Sim, K., & Hart, E. (2013, July). Generating single and multiple cooperative heuristics for the one dimensional bin packing problem using a single node genetic programming island model. In Proceedings of the 15th annual conference on Genetic and evolutionary computation (pp. 1549-1556).
[13] Buljubašić, M., & Vasquez, M. (2016). Consistent neighborhood search for one-dimensional bin packing and two-dimensional vector packing. Computers & Operations Research, 76, 12-21.
[14] Bhatia, A. K., & Basu, S. K. (2004, November). Packing bins using multi-chromosomal genetic representation and better-fit heuristic. In International Conference on Neural Information Processing (pp. 181-186). Springer, Berlin, Heidelberg..
[15] Levine, J., & Ducatelle, F. (2004). Ant colony optimization and local search for bin packing and cutting stock problems. Journal of the Operational Research society, 55(7), 705-716.
[16] Layeb, A., & Boussalia, S. R. (2012). A novel quantum inspired cuckoo search algorithm for bin packing problem. International Journal of Information Technology and Computer Science, 4(5), 58-67.
[17] Layeb, A., & Chenche, S. (2012). A novel grasp algorithm for solving the bin packing problem. International Journal of Information Engineering and Electronic Business, 4(2), 8.
[18] Dokeroglu, T., & Cosar, A. (2014). Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms. Computers & Industrial Engineering, 75, 176-186.
[19] Bayraktar, T., Aydin, M. E., & Dugenci, M. (2014). A memory-integrated artificial bee algorithm for 1-D bin packing problems. Proc. CIE IMSS, 1023-1034.
[20] Quiroz-Castellanos, M., Cruz-Reyes, L., Torres-Jimenez, J., Gómez, C., Huacuja, H. J. F., & Alvim, A. C. (2015). A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Computers & Operations Research, 55, 52-64.
[21] Kucukyilmaz, T., & Kiziloz, H. E. (2018). Cooperative parallel grouping genetic algorithm for the one-dimensional bin packing problem. Computers & Industrial Engineering, 125, 157-170.

[22] El-Ashmawi, W. H., & Abd Elminaam, D. S. (2019). A modified squirrel search algorithm based on improved best fit heuristic and operator strategy for bin packing problem. Applied Soft Computing, 82, 105565.
[23] Borgulya, I. (2021). A hybrid evolutionary algorithm for the offline Bin Packing Problem. Central European Journal of Operations Research, 29(2), 425-445.
[24] Hartono, N., Ismail, A. H., Zeybek, S., Caterino, M., Jiang, K., Sahin, M., & Natalia, C. (2022). The 1-D bin packing problem optimisation using bees algorithm. Journal Industrial Servicess, 7(2), 323-326.
[25] Brandao, F., & Pedroso, J. P. (2016). Bin packing and related problems: General arc-flow formulation with graph compression. Computers & Operations Research, 69, 56-67.
[26] Wei, L., Luo, Z., Baldacci, R., & Lim, A. (2020). A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. INFORMS Journal on Computing, 32(2), 428-443.
[27] Gilmore P. C., and Gomory R. E. 1963. "A Linear Programming Approach to the Cutting Stock Problem—Part II," no. Annals of Operations Research: 863–88. https://doi.org/https://doi.org/10.1287/opre.11.6.863.
[28] Schwerin P., Wäscher G. The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP. International Transactions in Operational Research, 4: 337-389, 1997.
[29] Schwerin P., Wäscher G. A new lower bound for the bin-packing problem and its integration to MTP. Pesquisa Operacional, 19: 111-129, 1999.
[30] Wäscher G., Gau T. "Heuristics for the one-dimensional cutting stock problem: A computational study". OR Spektrum, 18: 131-144, 1996.
[31] Schoenfield J. E. Fast, exact solution of open bin packing problems without linear programming. Draft, US Army Space & Missile Defense Command, Huntsville, Alabama, USA, 2002.