



On the Impact of Fuzzy Constraints in the Variable Size and Cost Bin Packing Problem

Jorge Herrera-Franklin¹ , Alejandro Rosete² , Milton García-Borroto² ,
Carlos Cruz-Corona³ , and David A. Pelta³

¹ Maritime Transportation Division, Center of Research and Environmental
Management of Transport, La Habana, Cuba
franklin@cimab.transnet.cu

² Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE),
La Habana, Cuba
{[rosete](mailto:rosete@ceis.cujae.edu.cu),[mgarciab](mailto:mgarciab@ceis.cujae.edu.cu)}@ceis.cujae.edu.cu

³ Department of Computer Science and AI, Universidad de Granada, Granada, Spain
{[carloscruz](mailto:carloscruz@decsai.ugr.es),[dpelta](mailto:dpelta@decsai.ugr.es)}@decsai.ugr.es
<http://www.cimab.transnet.cu>

Abstract. The Variable Size and Cost Bin Packing Problem (VSCBPP) consists of minimizing the cost of all bins used to pack a set of items without exceeding the bins capacities. It is a well known NP-Hard problem with many practical applications.

In this contribution we assume that the capacity of a bin can be understood in a flexible way (so it may allow some overload) thus leading to a fuzzy version of the VSCBPP with fuzzy constraints.

We solve the proposed fuzzy VSCBPP by using the parametric approach based on α -cuts, thus defining a set of related crisp problems.

By using three different solving algorithms and several instances, we explore the impact of different degrees of relaxation not only in terms of cost, but also in the structure of the solutions.

Keywords: Combinatorial optimization · Variable Size and Cost Bin Packing Problem · Fuzzy constraint · Parametric approach

1 Introduction

The Variable Sized Bin Packing Problem (VSBPP) is a generalization of the Bin Packing Problem that was first formalized by Friesen and Langston [5]. It consists in packing a set of items in a set of heterogeneous bins with different sizes or capacities. The objective is to minimize the number of bins that are used. For each size, it is assumed an inexhaustible supply of bins. Crainic *et al.* [2] states that by minimizing the cost of all used bins, the problem became the Variable Size and Cost Bin Packing Problem (VSCBPP).

Some variants of the problem were defined to allow the management of imprecise and/or uncertain information in the problem data. One of the pioneers treating this subject is Crainic *et. al.* [3] studying a real-life application in logistics with uncertainty on the characteristics of the items. Also Wang *et. al.* [16] describes a chance-constrained model where the item sizes are uncertain, while Peng and Zhang [10] introduce the uncertainty on item volumes and bin capacities.

Here, we consider that the capacity of a bin is associated with the maximum weight it can hold. So, for example bins with 25, 50 or 75 kg. capacity may exist. In the standard formulation of the problem, such values are used as crisp constraints. However, it has perfect sense to consider such capacity values as approximate ones and thus using fuzzy constraints instead of crisp ones. In other words, some overloading will be allowed in the bins. To the best of our knowledge, there are not variants of the Variable Size and Cost Bin Packing Problem (VSCBPP) with fuzzy constraints.

The aim of the paper is twofold. Firstly, we introduce a fuzzy version of the VSCBPP that allows some overloading of the bins, which means to relax the satisfaction of capacity constraints; and secondly, we explore the impact that the fuzzy constraints (and the associated relaxations) have, not only in terms of cost, but also in the structure of the solutions. In order to do this, some randomly generated instances of the proposed fuzzy VSCBPP (FVSCBPP) are solved following the parametric approach that transforms the fuzzy problem into a set of crisp problem based on α -cuts [4, 14]. Then, each of these instances is solved by an exact solver (SCIP) [15] and two problem-specific heuristics.

The paper is organized as follows. Section 2 presents the VSCBPP fuzzy model. Section 3 explains how the proposed FVSCBPP may be solved by using a parametric approach. This is illustrated in several instances presented in Sect. 4. Finally, Sect. 5, is devoted to conclusions.

2 Fuzzy Variable Size and Cost Bin Packing Problem (FVSCBPP)

In this section we firstly present the basic Variable Size and Cost Bin Packing Problem (VSCBPP) before introducing the proposed fuzzy extension. Problem parameters and the standard VSCBPP formulation [1, 7, 8] are described next. Being

| | |
|--------------------------------|---|
| $I = \{1 \dots, i, \dots, n\}$ | set of items |
| w_i | weight of the item $i \in I$ |
| $J = \{1 \dots, j, \dots, m\}$ | set of bins |
| W_j | capacity (or size) of the bin $j \in J$ |
| C_j | cost of the bin $j \in J$ |
| x_{ij} | binary variable: 1 if item i is packed in bin j ; 0 otherwise |
| y_j | binary variable: 1 if bin j is used; 0 otherwise |

Then, the VSCBPP is then formulated as follows:

$$\text{Min} \sum_{j \in J} C_j y_j \quad (1)$$

$$\text{s.t.} \sum_{j \in J} x_{ij} = 1, \quad i \in I \quad (2)$$

$$\sum_{i \in I} w_i x_{ij} \leq W_j y_j, \quad j \in J \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad i \in I, \quad j \in J \quad (4)$$

$$y_j \in \{0, 1\}, \quad j \in J \quad (5)$$

The objective function (1) minimizes the cost of the bins used for packing all the items. Constraint (2) ensures that each item i is packed in one and only one bin (items are not divisible). Inequality (3) is the capacity constraint: for each used bin j , the sum of weights of packed items can not exceed its capacity; (4) and (5) are domain constraints. This formulation involves every single bin regardless its type, i.e., a list of bins is one of the problem inputs and it may have more bins than items since there must be enough bins to pack all the items that fulfill the constraint for every type (3). Here, the term type is referred to the capacity of the bin, i.e., its size.

Here we consider that the capacity constraint (3) can be understood in “flexible” (fuzzy) terms:

$$\sum_{i \in I} w_i x_{ij} \leq^f W_j y_j, \quad j \in J \quad (6)$$

where \leq^f stands for “approximately smaller than or equal to”.

This implies that solutions may be either feasible or infeasible depending on the interpretation of the fuzzy relation (6). Indeed, all solutions may be considered feasible with different degrees of membership.

A decision maker may clearly states that the solutions that do not exceed the bin capacity W_j are definitely feasible. In addition, some small overloads may be also considered feasible. Let's suppose there is a bin j with capacity $W_j = 10$. A solution where the items in j weights 10.01 units will have a higher degree of feasibility than another one with weight 12. In turn, a solution that try to pack items with weight 20 in such bin with 10 units of capacity, may be consider unfeasible. In terms of decision making, this relaxation that may imply a small overload in the bins may be preferable if it allows to reduce the total cost in the objective function (1).

To model this situation, we consider that a decision maker must define the tolerance T_j that defines the maximum admissible relaxation for each bin j . Figure 1 shows the function to measure the degree of accomplishment (for a given solution) of constraint (6) in terms of the bin capacity W_j and tolerance T_j .

To understand such function, let's call tw_j to the sum of the weights of the items placed in each bin j . Then, if $tw_j \leq W_j$, then such assignment of items

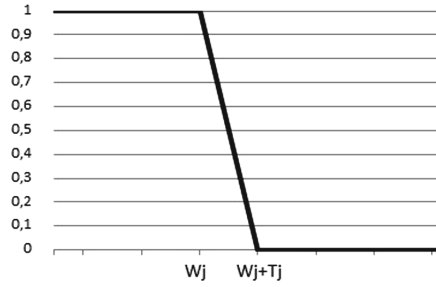


Fig. 1. Membership function for the accomplishment of capacity constraint (y axis) in terms of the sum of the items' weights (x axis).

to the bin is feasible with degree 1. In turn, if $tw_j > W_j + T_j$ feasibility is zero. When $tw_j \in [W_j, W_j + T_j]$ then such solution will be feasible with different degrees between $[0, 1]$.

In order to solve the problem, the fuzzy constraint will be managed using the parametric approach [14] and the concept of α -cuts. In very simple terms, this allows to obtain several crisp instances based on different values of α . If S is the whole set of solutions we may define S^α as the crisp set of solutions that satisfy the constraints with, at least, a given degree α .

$$S^\alpha = \{s \in S \mid \mu(s_j) \geq \alpha\} \quad (7)$$

According to (7), a solution $s \in S$ is considered α -feasible if it is feasible with a degree above α . This implies that there are different sets S^α of feasible solutions for different values of α . As different sets S^α may include different sets of solutions, the optimal solution for each S^α may be different. So to solve the fuzzy version of VSCBPP we will use the constraint (8) instead of (3).

$$\sum_{i \in I} w_i x_{ij} \leq (W_j + (1 - \alpha)T_j)y_j, j \in J \quad (8)$$

When $\alpha = 1$ the most restricted definition of capacity is taken into account, thus having the original crisp problem (no flexibility). When $\alpha = 0$, the most flexible situation is reached. The allowed bin overload is maximum. Consequently, the value α defines the degree of relaxation that is admitted.

3 Solving the Fuzzy VSCBPP

The parametric approach [14], illustrated in Fig. 2, is used to solve the problem. The main idea is to transform the fuzzy problem into a family of crisp problems. Initially, a set of different α values is defined and for each value, a crisp problem is obtained and then solved. Finally, it should be remarked that the final result consists of a *set* of solutions related to each value of α .

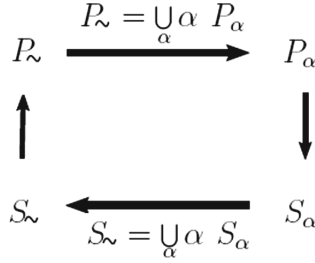


Fig. 2. General scheme of the parametric approach

In our case, we consider 11 values of $\alpha \in \{0.0, 0.1, \dots, 0.9, 1.0\}$ ¹. For solving the different crisp problems we consider in this paper three solution methods. An exact solver based on Integer Linear Programming (SCIP solver), and two heuristics: First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) [6, 11]. FFD is a deterministic heuristic that place each item in the first bin where it is possible to place it, where BFD chooses the bin where the item best fits. Both heuristics repeat this process item by item, taking them in descending order. It is worth noting that heuristics methods do not guarantee optimality.

In the case of the SCIP solver, the optimality is only guaranteed after a considerable amount of time. In controlled conditions (for example, a maximum execution time of one hour for each instance) we observed that optimality is not guaranteed for all cases. It is import to remark that our focus here is on analyzing each fuzzy solution (i.e., the set of solutions obtained for the base instance with different values of α).

3.1 Test Instances

In this experiment we take three base instances following previous works in VSCBPP [1, 8]. These base instances are used as the crisp original instances, associated with $\alpha = 1$. These three base instances of the FVSCBPP result in 33 crisp ones that need to be solved (one for each value of the 11 α values considered). Each base instance contains 25 items and 3 bin types with $W \in \{50, 100, 150\}$, and the tolerance was set to $T \in \{6, 5, 7\}$, respectively. The weight of each item is randomly assigned using a uniform distribution in [20, 120].

It must be remarked that the three base instances differ in the relation between the cost C_j of each bin and its capacity W_j , as it was previously conceived in other works [1, 7]. This functional relation is the origin of the name used to identify each instance: Concave (Cc) where $C_j = 15\sqrt{W_j}$, Linear (Ln) where $C_j = 10W_j + 32$, and Proportional (Pr) where $C_j = 0.1W_j$.

The three cost functions (Cc, Ln, and Pr) produce different behaviors. For example, according to the function Pr is the same to use three bins with $W_j = 50$ than using one bin with $W_j = 150$, and both are the same than using a bin with

¹ Alternative schemes for exploring the values of α were recently presented in [13].

$W_j = 50$ plus another with $W_j = 100$. In the other cost functions, it is better to use a bin with $W_j = 150$ than using three bins with $W_j = 50$ (in the case of the function Ln this implies 5% additional cost, while in Cc this value is 73%). The same occurs in the comparison of using a bin of $W_j = 150$ with respect to use a bin $W_j = 50$ and a bin with $W_j = 100$ (in the case of Ln this implies 4% additional cost, while in Cc this value is 38%). These are just some examples of the implications of the cost function.

The test instances are available in www.cimab.transnet.cu/files/iFSCBPP.zip in order to allow replication of the presented results.

4 Results and Discussion

As stated before, for each test instance we define 11 α values: $\alpha \in [0.0, 0.1, 0.2, \dots, 1.0]$, leading to 33 crisp problems. Each one of these problems is solved in two ways. Firstly using a Mixed Integer Programming solver SCIP [12], with the problem model coded using the ZIMPL format [9]. SCIP is expected to return the optimal solution. Secondly, two problem-specific heuristics for the Bin Packing Problem are used: First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) [6, 11].

Runs were performed in an Intel Core i5 processor with 2.4 GHz of clock speed and 8 Gb of RAM. The running time of both heuristics is less than one second (after the items are sorted). But for running the SCIP solver, a maximum number of 4 parallel threads and 60 min (one hour) of maximal execution time were set.

The analysis of the results is divided in two parts. In the first one we consider the impact of α in the solutions' costs. Then, we analyze such impact in terms of the solutions' structure.

Figure 3 shows the impact of the α values on the solutions costs for every solver and test instances. A clear tendency is observed: as the problem is more relaxed ($\alpha \rightarrow 0$), the cost of the solution diminished. When α is near to 0 (fully relaxed case), the available space in the bins is higher and less cost is needed to store all the items. As the relaxation decreases ($\alpha \rightarrow 1$) the cost increases.

Figure 3 shows an additional interesting feature. When a heuristic is used, there is no guarantee to obtain a better solution if the problem is more relaxed. This is not the case with the SCIP solver, where if $\alpha_1 < \alpha_2$ then the corresponding solutions s_1, s_2 satisfies that $f(s_1) \leq f(s_2)$.

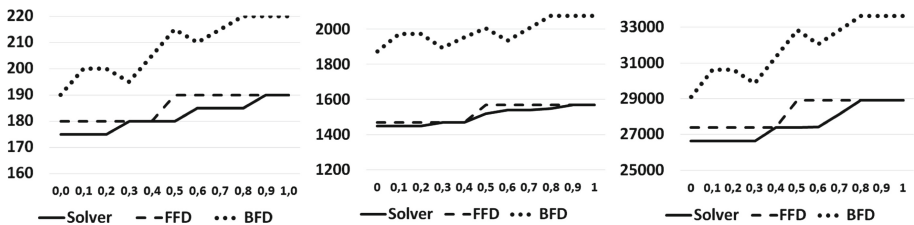


Fig. 3. Impact of the constraint violation in the cost (α in x-axis vs. cost in y-axis) for the test instances: Pr (left), Cc (center), Ln (right).

Finally, we can observe that the heuristic FFD obtains very similar results to those of the SCIP solver but using a very simple approach and an extremely reduced time. Although for most of the cases the SCIP solver required less than a minute, for some cases (mainly in the Ln instance) it did not finish within an hour. In such a case, the best solution found up to that point is reported.

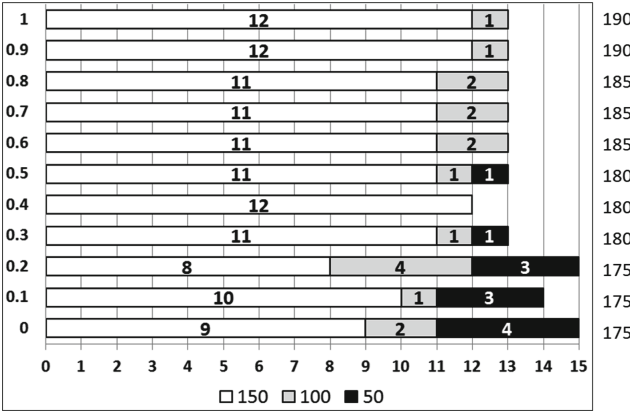


Fig. 4. Structure of the solutions (SCIP Solver) for the Pr instance in terms of α (y-axis). The number of bins of each type is shown.

In order to analyze the impact of α in terms of solutions' structure, Fig. 4 shows the solutions obtained by the SCIP solver for the Pr instance for each α value. Every row displays the number of bins of each type (150: white, 100: grey, 50: black) used together with the cost.

If the capacity constraint is very strict, 12 big bins and 1 middle sized are needed. As the relaxation increases, a better cost can be achieved with 11 big bins and 2 middle sized. The cases of $\alpha = 0.5$ and $\alpha = 0.4$ are also interesting. In the former, the three types of bins are used in a solution with cost 180. But an additional relaxation of the constraint allows to pack all the items using 12 big bins. A similar situation happens for the most relaxed cases $\alpha \geq 0.2$ where three different solutions with the same cost are displayed. So we have here another benefit of using fuzzy constraints and the parametric approach, where in a simple way we can obtain different design decisions (solutions) that provide the decision maker with a richer information beyond the cost. A similar analysis can be done for the other instances (Ln and Cc) based on Fig. 5 and Fig. 6.

The last analysis aims to observe how the capacity constraint is violated. It is important to note that having the opportunity to violate the capacity of the bins, does not mean that all the bins are overloaded.

Figure 7 shows the solutions obtained by every algorithm in the two extreme situations: $\alpha = 1$ and $\alpha = 0$. White bars correspond to big bins (capacity

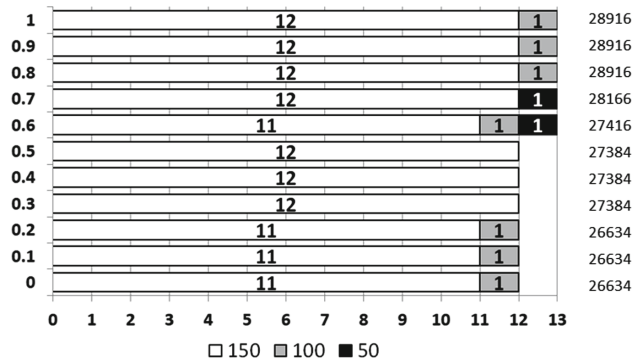


Fig. 5. Structure of the solutions (SCIP Solver) for the Ln instance in terms of α (y-axis). The number of bins of each type is shown.

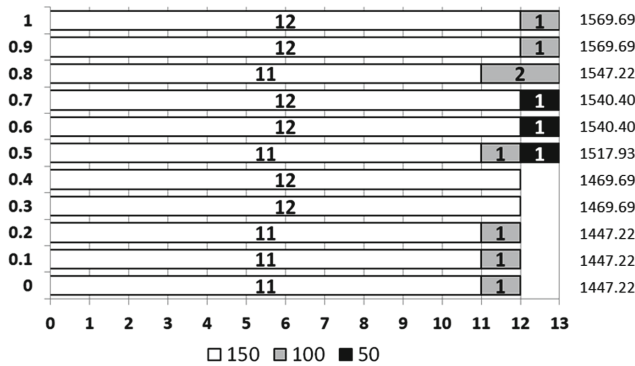


Fig. 6. Structure of the solutions (SCIP Solver) for the Cc instance in terms of α (y-axis). The number of bins of each type is shown.

150), grey ones to mid-size bins (capacity 100) and black ones to small size bins (capacity 50). If a bar is taller than its capacity, then such bin are overloaded (makes use of the relaxation). The horizontal lines within the bars identify the items packed.

Figure 7 (on top) displays the crisp case (no relaxation is allowed). Both SCIP solver and FFD heuristic achieved a solution with the same cost and the same structure. They used 12 big bins and 1 middle sized but, as it can be observed the items are packed differently. Again, this kind of visualization allows a user to take a more informed decision. One may argue that aspects like the level of occupancy of the bins should be taken into account within the objective function. However, in our opinion, that would complicate the solution of the problem.

If we consider the fully relaxed version ($\alpha = 0$), we observe that the capacity violation is small. Nevertheless, it allows for a relevant cost reduction. If the comparison is made in a column wise manner, we can compare the none relaxed vs. the fully relaxed solutions obtained by every algorithm. The SCIP solver

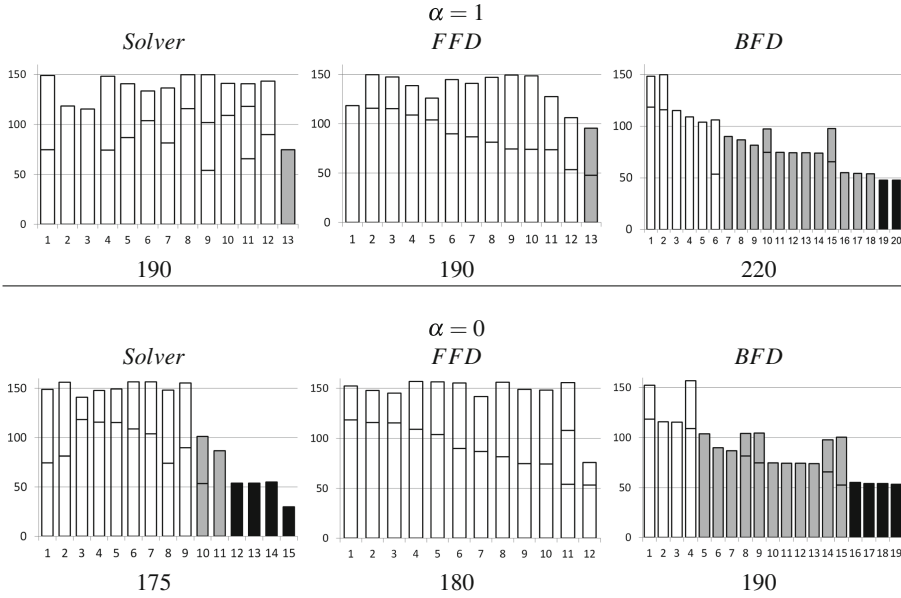


Fig. 7. Structure of the solutions for Cc instance obtained by every algorithm with no relaxation ($\alpha = 1$) and with the fully relaxed condition ($\alpha = 0$).

reduces the cost in 15 units by decreasing the usage of big bins while adding more of the smaller ones. In turn FFD, produces a cheaper solution differently: use big bins with a slightly larger capacity (taking profit of the relaxation). Finally, BFD obtained the greatest decrease (30 units) but using 30 bins. It can be noticed that such solution has the same cost of those obtained by SCIP and FFD when $\alpha = 1$. Despite the slight violation of the capacity, it is clear that the BFD solution may be harder to “manage”: larger number of bins and use the three available types.

5 Conclusions

This paper presents a fuzzy version of the Variable Size and Cost Bin Packing Problem (VSCBPP), where the bins capacity is considered flexible. Allowing such flexibility is relevant in many practical situations because it may allow to obtain cheaper solutions.

The proposed fuzzy version of VSCBPP is expressed in terms of fuzzy constraints that allow to respect the limitations regarding the capacities with a certain tolerance. Based on the parametric approach, the solution of the fuzzy problem consists of a set of solutions that may show a trade-off between relaxation (violation of the original condition) and benefit (cost reduction).

Our experimental study shows first that all the algorithms tested can achieve cheaper solutions as the relaxation increases. The analysis of the solution

revealed that different algorithms manage the flexibility in different ways, thus allowing to obtain a diverse set of solutions.

This is a crucial aspect for a decision maker, for whom, different solutions with similar cost can be provided. Then, such solutions can be analyzed from other points of view beyond costs like how easy/hard is to manage the selected bins, or how easy/hard is to transport them. Although such features may be included in the cost function (this is far from trivial), this will lead to a more complex and harder to solve model.

Acknowledgements. This work is supported by the project “*Optimization of the distribution and composition of the fleet in midterm considering acquisition mode*” of the Maritime Transport Division of the Center of Research and Environmental Management of Transport (Cimab) and the PhD Academic Program of the Informatics Engineering Faculty of the Universidad Tecnológica de La Habana “José Antonio Echeverría” (CUJAE) as part of PhD research of the first author. D. Pelta and C. Cruz acknowledge the support of Project- TIN2017-86647-P from Spanish Ministry of Economy and Competitiveness, (including FEDER funds, from the European Union).

References

1. Correia, I., Gouveia, L., da Gama, F.S.: Solving the variable size bin packing problem with discretized formulations. *Comput. Oper. Res.* **35**(6), 2103–2113 (2008). <https://doi.org/10.1016/j.cor.2006.10.014>. ISSN 0305-0548
2. Crainic, T.G., Perboli, G., Rei, W., Tadei, R.: Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Comput. OR* **38**, 1474–1482 (2011)
3. Crainic, T.G., Gobbato, L., Perboli, G., Rei, W., Watson, J.P., Woodruff, D.L.: Bin packing problems with uncertainty on item characteristics: an application to capacity planning in logistics. *Procedia - Soc. Behav. Sci.* **111**(5), 654–662 (2014). <https://doi.org/10.1016/j.sbspro.2014.01.099>
4. Ebrahimnejad, A., Verdegay, J.L.: *Fuzzy Sets-Based Methods and Techniques for Modern Analytics*. SFSC, vol. 364. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-73903-8>
5. Friesen, D.K., Langston, M.A.: Variable sized bin-packing. *SIAM J. Comput.* **15**, 1 (1986). <https://doi.org/10.1137/0215016>. ISSN 1095-7111
6. Haouari, M., Serairi, M.: Heuristics for the variable sized bin-packing problem. *Comput. Oper. Res.* **36**, 2877–2884 (2009). <https://doi.org/10.1016/j.cor.2008.12.016>. ISSN 0305-0548
7. Hemmelmayr, V., Schmid, V., Blum, C.: Variable neighbourhood search for the variable sized bin packing problem. *Comput. Oper. Res.* **39**, 1097–1108 (2012). <https://doi.org/10.1016/j.cor.2011.07.003>. ISSN 0305-0548
8. Kang, J., Park, S.: Algorithms for the variable sized bin packing problem. *Eur. J. Oper. Res.* **147**, 365–372 (2003). 0377–2217/03, ISSN 0377–2217
9. Koch, T.: *Zimpl User Guide* (Zuse Institute Mathematical Programming Language)). PH.D. thesis, Technische Universität Berlin, February 2004. <http://www.zib.de/Publications/abstracts/ZR-04-58/>. zIB-Report 04–58

10. Peng, J., Zhang, B.: Bin packing problem with uncertain volumes and capacities (2012). http://www.researchgate.net/profile/Jin_Peng13/publication/265533462_Bin_Packing_Problem_with_Uncertain_Volumes_and_Capacities/links/55657a3808aec4b0f4859d5d.pdf
11. Pillay, N., Qu, R.: Hyper-heuristics: theory and applications. In: *Packing Problems, Natural Computing Series*, pp. 67–73. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-96514-7>
12. Schwarz, C.: An Introduction to SCIP. University of Bayreuth, September 2010
13. Torres, M., Pelta, D.A., Lamata, M.T.: A new approach for solving personalized routing problems with fuzzy constraints. In: *IEEE International Conference on Fuzzy Systems (FUZZ)* (2018)
14. Verdegay, J.L.: Fuzzy mathematical programming. In: *Fuzzy Information and Decision Processes*, pp. 231–237 (1982)
15. Vigerske, S., Gleixner, A.: SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optim. Methods Soft.* **33**(3), 563–593 (2018). <https://doi.org/10.1080/10556788.2017.1335312>
16. Wang, S., Li, J., Mehrotra, S.: Chance-constrained bin packing problem with an application to operating room planning (2019). <http://www.optimization-online.org/DB.HTML/2019/02/7053.html>