Solving the Variable Size Bin Packing Problem with Discretized Formulations

Isabel Correia[1], Luís Gouveia[2*] and Francisco Saldanha-da-Gama[2]

[1] Department of Mathematics / Centre of Mathematics and Applications
Faculty of Science and Technology, New University of Lisbon
Quinta da Torre, 2829-516 Monte da Caparica, Portugal
isc@fct.unl.pt

[2] Department of Statistics and Operational Research/Operational Research Centre
Faculty of Science, University of Lisbon
Bloco C6, Piso 4, 1749-016 Lisboa, Portugal
legouveia@fc.ul.pt, fsgama@fc.ul.pt

Abstract

In this paper we study the Variable Size Bin Packing Problem (VSBPP) which is a generalization of the Bin Packing Problem where bins of different capacities (and different costs) are available for packing a set of items. The objective is to pack all the items minimizing the total cost associated with the bins. We discuss applications of the VSBPP and propose and discuss one generic (non-linear integer programming) formulation as well as two linear integer programming formulations. One of these formulations overcomes the non-linearity of the original model by simply adding explicitly the class of the bin used. The other, less straightforward, uses a so-called discretized model reformulation technique already proposed for other problems (see Gouveia (1995) and Gouveia and Saldanha da Gama (2006)), and shows that we need not use explicitly the information on the type of bin used provided we know the amount packed in it. These two models are, then, compared in terms of the linear relaxation bounds. New valid inequalities suggested by the decision variables of the discretized models are also proposed to strengthen the original linear relaxation bounds. Computational results are presented showing that the valid inequalities proposed not only enhance the linear programming relaxation bound but may also be extremely helpful when using a commercial package for solving the VSBPP to optimality.

Keywords: Variable Size Bin Packing, Reformulations, Extended formulations

* Corresponding author

1. Introduction

In this paper we study the Variable Size Bin Packing Problem (VSBPP). Given a set of items, each one with a certain weight, and a set of bins partitioned into several subsets, each subset corresponding to bins of a given positive capacity and a given cost, the objective is to pack all the items in the bins and minimizing the total cost associated with the chosen bins. It is assumed that for each class there are an unlimited number of bins. This variation of the classic bin packing problem (which only has one class of bins) has many practical applications:

i) Loading truck problems when weight is the only 'dimension' to be considered, considers the situation where more than one truck of each size / weight limit is available and the goal is to minimize the total cost associated with the selected trucks.

ii) Cutting-stock problems. The bins are associated to standard lengths of some material like paper, wood, or electric cables from which 'items' must be cut. In this case, the items refer to specific dimensions that have to be cut from the standard lengths. When, more than one standard length is available we obtain the VSBPP.

iii) In machine scheduling, the VSBPP arises when a set of tasks/jobs with known arbitrary processing times need to be executed and different classes of processors are available to parallelize the processing. In this case the goal is (typically) to schedule all the tasks minimizing a cost associated with the processors.

The literature on the MBBP is scarce. Friesen and Langston (1986) have studied this problem with the objective of minimizing the total capacity associated with the bins used (which is equivalent to minimizing the total cost associated with the bins if this cost is proportional to the size of the bins). The authors propose several approximation schemes giving asymptotic worst case bounds. Kang and Park (2003) propose two greedy algorithms for solving three particular cases of the problem. These cases are defined in terms of the divisibility of the weights of the items and/or the capacities of the bins.

The VSBPP contains the classical one-dimensional bin packing problem (BPP) as a particular case which is known to be a NP-Hard problem (Garey and Johnson (1979)). Thus, the VSBPP is NP-Hard as well.

In this paper we propose and analyze several formulations for VSBPP. We start by presenting a generic formulation with a general cost function (Section 2). The objective function of this model is general enough to capture non-linearities that may exist in the costs associated with the use of the bins. To overcome the non-linearity of the objective function we consider two alternative decision variable sets that lead to two alternative integer linear programming formulations. One of these models is straightforward, and solves the non-linearity of the model above by simply adding to the variable definition information on the class of the bin used. The other, less straightforward, uses a so-called discretized model reformulation technique already proposed for other problems (see Gouveia (1995) and Gouveia and Saldanha da Gama (2006)), and shows that we need not use explicitly the information on the type of bin used provided we know the amount packed in it. These two models (see Section 3) are, then, compared in terms of the linear relaxation bounds. New valid inequalities suggested by the decision variables of the discretized models are also proposed to strengthen the original linear relaxation bounds. Computational results (see Section 4) are presented showing that the valid inequalities proposed not only enhance the linear programming relaxation bound but may also be extremely helpful when using a commercial package for solving optimally VSBPP.

For any model P we denote by $v(P)$ its optimal value and let $v(\overline{P})$ denote the optimal value of the corresponding linear programming relaxation.

2. Generic Formulation

Let J denote the set of items with integer weight w_j ($j \in J$) and I denote the set of bins. Here, we consider the case where the set I is partitioned into subsets I_k , ($k \in K$), each subset corresponds to bins of a given positive capacity b_k and a given cost f_k. Without loss of generality we will also assume that $w_1 \leq w_2 \leq \ldots \leq w_n$ (n = |J| = $|I_k|$, k=1,…,|K|), $b_1 \leq b_2 \leq \ldots \leq b_m$ (m = |K|), $w_j \leq b_m$ ($j \in J$).

Consider the following generic formulation for the problem (we consider any ordering of the bins) involving binary variables x_{ij} indicating whether item j is packed into the i^{th} bin ($i \in I$, $j \in J$) and integer variables v_i representing the amount stored in bin i ($i \in I$). Here we obviously, assume that the capacity of the i^{th} bin is enough for packing the amount equal to v_i:

$$\text{(P1)} \quad \text{Min} \quad \sum_{i=1}^{n} f(v_i) \qquad (1)$$

$$\text{s. to} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$\sum_{j=1}^{n} w_j x_{ij} = v_i \qquad i \in I \qquad (3)$$

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

$$v_i \geq 0 \qquad i \in I \qquad (5)$$

$$v_i \leq b_m \qquad i \in I \qquad (6)$$

where

$$f(v_i) = \begin{cases} f_1 & if \quad 1 \leq v_i \leq b_1 \\ \vdots & \vdots \quad \vdots \\ f_k & if \quad b_{k-1} + 1 \leq v_i \leq b_k \\ \vdots & \vdots \quad \vdots \\ f_m & if \quad b_{m-1} + 1 \leq v_i \leq b_m \end{cases} .$$

The objective function (1) minimizes the cost of the bins used for packing all the items. Note the non-linearity of this objective function. The value $f(v_i)$ gives the cost of the minimum capacity bin which can pack the amount v_i. Constraints (2) ensure that each item j is packed; equalities (3) state the amount packed in each bin i is equal to the amount v_i; constraints (4) are the integrality constraints, whereas (5) and (6) establish bounds for the amount packed in each bin i.

3. Integer Linear Programming Models

In this next section we discuss two mixed integer linear programming models. One is straightforward, and solves the non-linearity of the model above by simply adding explicitly the class of the bin used, to the information on the i^{th} bin used. The other, less straightforward, uses a so-called discretized model reformulation technique already proposed for other problems (see Gouveia (1995) and Gouveia and Saldanha da Gama (2006)), and shows that we need not use explicitly the information on the type of bin used provided we know the amount packed in it.

3.1 Model P2

By using additional binary variables y_{ik} indicating whether the i^{th} bin has capacity b_k and is used in the solution, we can write the following formulation for the VSBPP:

$$(P2) \quad \text{Min} \quad \sum_{i=1}^{n}\sum_{k=1}^{m} f_k y_{ik} \qquad (7)$$

$$\text{s. to} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$\sum_{j=1}^{n} w_j x_{ij} = v_i \qquad i \in I \qquad (3)$$

$$v_i \le \sum_{k=1}^{m} b_k y_{ik} \qquad i \in I \qquad (8)$$

$$\sum_{k=1}^{m} y_{ik} \le 1 \qquad i \in I \qquad (9)$$

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

$$v_i \ge 0 \qquad i \in I \qquad (5)$$

$$y_{ik} \in \{0,1\} \qquad i \in I,\ k \in K \qquad (10)$$

The "new" inequalities (9) are consistency constraints guaranteeing that each bin only has one of the given capacity values. The new capacity constraints (8) link the v_i with the y_{ik} variables and together with (9), they guarantee that the bin chosen to pack the amount selected for the i^{th} bin has enough capacity. The new constraints (10) are domain constraints. Constraints (3) permit us to eliminate the v_i variables from the model leading to the following model *P3*:

$$(P3) \quad \text{Min} \quad \sum_{i=1}^{n}\sum_{k=1}^{m} f_k y_{ik} \qquad (7)$$

s. to $\sum_{i=1}^{n} x_{ij} = 1 \qquad j \in J \qquad (2)$

$$\sum_{j=1}^{n} w_j x_{ij} \le \sum_{k=1}^{m} b_k y_{ik} \qquad i \in I \qquad (11)$$

$$\sum_{k=1}^{m} y_{ik} \le 1 \qquad i \in I \qquad (9)$$

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

$$y_{ik} \in \{0,1\} \qquad i \in I,\ k \in K \qquad (10)$$

The following result is a straightforward adaptation of a similar result presented for the BPP by Martello and Toth (1990) when the costs satisfy economies of scale or are proportional to the capacities.

Result 1: An optimal solution for $\overline{P3}$ is given by

$$x_{ii} = 1,\ x_{ij} = 0\ \ (i \ne j),\ y_{im} = w_i / b_m,\ \ y_{ik} = 0\ \ (k \ne m).$$

Proof: The minimum space required to pack all the items is given by $\sum_{j=1}^{n} w_j$. The lowest cost for packing all items is attained when all the items are packed in the bins for which the unitary cost is the smaller. Under economies of scale or in the situation where the costs are proportional to capacities, a bin with the largest capacity corresponds to the lowest unitary cost, which is given by f_m / b_m. It is easy to verify that this solution is feasible for $\overline{P3}$ and has cost given by $f_m / b_m \times \sum_{j=1}^{n} w_j$. ▲

This result also indicates that the bound $v(\overline{P3})$ needs to be improved. In an attempt to strengthen this lower bound, we present, next, two classes of valid inequalities for the problem. Both of them are similar in flavour to inequalities used for improving the linear programming relaxation of discrete location problems (see, for instance Cornuejols *et al* (1991) and Gouveia and Saldanha-da-Gama (2006)). The first inequality

$$\sum_{i=1}^{n} \sum_{k=1}^{m} y_{ik} \ge \left\lceil \frac{\sum_{j=1}^{n} w_j}{b_m} \right\rceil. \qquad (12)$$

states that we need to select enough bins (and with enough capacity) to pack all the given items. The inequalities in the second set are as follows

$$x_{ij} \le \sum_{k=k'}^{m} y_{ik} \qquad i \in I,\ j \in J \qquad (13)$$

where $k' = \min\{k : b_k \ge w_j\}$ (that is, *k'* represents the index of the smallest bin size that is able to pack item *j*).

3. 2. The Discretized Model

A different way to overcome the non-linearity of the objective function of *P1* is to make use of a so-called discretized model (see, for instance, Gouveia and Saldanha-da-Gama (2006)). The main idea of using such a type of model is to use "new" binary variables z_i^q $(i \in I, q \in \{1,...,b_m\})$ indicating whether the i^{th} bin $(i \in I)$ is storing an amount equal to *q*. The interest about this discrimination technique is that we can obtain the discretized model from the original model *P1* by using the following set of linking constraints, to replace old with new variables:

$$v_i = \sum_{q=1}^{b_m} q z_i^q \qquad i \in I \qquad (14)$$

The discretized model becomes as follows:

$$(DM) \quad \text{Min} \quad \sum_{i=1}^{n} \sum_{q=1}^{b_m} f^q z_i^q \qquad (15)$$

$$\text{s. to} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$\sum_{q=1}^{b_m} z_i^q \le 1 \qquad i \in I \qquad (16)$$

$$\sum_{j=1}^{n} w_j x_{ij} = \sum_{q=1}^{b_m} q z_i^q \qquad i \in I \qquad (17)$$

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

$$z_i^q \in \{0,1\} \qquad i \in I,\ q \in \{1,...,b_m\} \qquad (18)$$

Inequalities (16) are consistency constraints for the new variables and state that if $z_i^q = 1$ for a given *i* and *q*, then $z_i^p = 0$ for all $(p \neq q)$. Constraints (17) are the new constraints obtained by the discretization technique and guarantee that the index *q* of the discretized variable associated equal to 1 and associated with bin *i* must be equal to the total weight of the items stored in *i*. The advantage of this model is obvious since by defining

$$f^q = \begin{cases} f_1 & if \quad 1 \leq q \leq b_1 \\ \vdots & \vdots \quad \quad \vdots \\ f_k & if \quad b_{k-1}+1 \leq q \leq b_k \\ \vdots & \vdots \quad \quad \vdots \\ f_m & if \quad b_{m-1}+1 \leq q \leq b_m \end{cases} .$$

we can write the model with a linear objective function (15).

3.3 Comparing the Two Models

In this section we show that the linear programming relaxation of the two models, *P3* and *DM* are equivalent. To show this result, consider the following set of equalities that relate the new sets of variables in the two previous models:

$$y_{ik} = \sum_{q=b_{k-1}+1}^{b_k} z_i^q \quad i \in I,\ k \in K,\ b_0 = 0 \qquad (19)$$

We shall use these equalities to show how to transform a solution that is feasible or optimal for one model to another that is feasible for the other and which has the same cost.

Result 2: $v(\overline{P3}) = v(\overline{DM})$

Proof: i) Let $\{x_{ij}, z_i^q\}$ be feasible for $\overline{DM}$ (here we do not require the solution to be optimal). Then, the solution $\{x_{ij}, y_{ik}\}$ with y_{ik} given by (19) is feasible for $\overline{P3}$. In fact, the validity of (2), (9) and non-negativity of variables x_{ij} and y_{ik} is straightforward. From (17) and using (19) we obtain

$$\sum_{j=1}^{n} w_j x_{ij} = \sum_{q=1}^{b_m} q z_i^q = \sum_{q=1}^{b_1} q z_i^q + \sum_{q=b_1+1}^{b_2} q z_i^q + \ldots + \sum_{q=b_{m-1}+1}^{b_m} q z_i^q \le$$

$$\le \sum_{q=1}^{b_1} b_1 z_i^q + \sum_{q=b_1+1}^{b_2} b_2 z_i^q + \ldots + \sum_{q=b_{m-1}+1}^{b_m} b_m z_i^q = b_1 \sum_{q=1}^{b_1} z_i^q + b_2 \sum_{q=b_1+1}^{b_2} z_i^q + \ldots + b_m \sum_{q=b_{m-1}+1}^{b_m} z_i^q =$$

$$= b_1 y_{i1} + b_2 y_{i2} + \ldots + b_m y_{im} = \sum_{k=1}^{m} b_k y_{ik}$$

This shows that (11) is satisfied. Clearly the cost of the solution $\{ x_{ij}, z_i^q \}$ in $\overline{DM}$ is the same as the cost of solution $\{ x_{ij}, y_{ik} \}$ in $\overline{P3}$.

This shows that $v(\overline{P3}) \le v(\overline{DM})$.

ii) Let $\{ x_{ij}^*, y_{ik}^* \}$ be an optimal solution for $\overline{P3}$. Clearly, constraints (11) are satisfied as equalities. Furthermore, we can assume that there exists at least one variable y_{ik} such that $y_{ik}^* > 0$. For each $l = 1,\ldots,m$ such that $y_{il}^* > 0$ define $z_i^{b_l} = y_{il}^*$ and $z_i^q = 0$ for $q = b_{l-1}+1,\ldots,b_l - 1$. For each $l = 1,\ldots,m$ such that $y_{il}^* = 0$ define $z_i^q = 0$ for $q = b_{l-1}+1,\ldots,b_l$. The solution $\{ x_{ij}^*, z_i^q \}$ with z_i^q defined as above is a feasible solution for $\overline{DM}$ with the same cost as $\{ x_{ij}^*, y_{ik}^* \}$ in $\overline{P3}$. In fact, (2), (16), and non-negativity of variables x_{ij} and z_i^q are clearly satisfied. Considering (11) for each $i \in I$ we have

$$\sum_{j=1}^{n} w_j x_{ij}^* = \sum_{k=1}^{m} b_k y_{ik}^* = \sum_{k: y_{ik}^* > 0} b_k y_{ik}^* = \sum_{k: y_{ik}^* > 0} b_k z_i^{b_k} = \sum_{q=1}^{b_m} q z_i^q$$

which is equivalent to (17).

Thus, we have shown that $v(\overline{P3}) \ge v(\overline{DM})$ and together with i) we conclude that $v(\overline{P3}) = v(\overline{DM})$. ▲

Apart from the domain constraints, models *P3* and *DM* have exactly the same number of constraints as well as the same number of x_{ij} variables. Excluding the situation where $b_m = \#K$, *DM* has, in general, much more binary variables than

P3.Thus, *DM* appears to be worse (from a computational point of view) than *P3* due to the large number of binary variables. Nevertheless, as our computational experience in section 4 will show, the model *DM* is worth considering for two reasons. First, we can eliminate quite straightforwardly many of the variables of the model. Second, the information attached to the discretized variables permit us to derive new and very intuitive inequalities that strongly enhance the original linear relaxation bound. Furthermore, any inequality that is defined in the variables of the non-discretized models *P2* and *P3* can be simply translated to *DM* using (14) and (19).

3.4 Reducing the Number of Variables of the DM Model

As observed above, the major drawback of model *DM* appears to be the large number of z_i^q variables. This number can be decreased in an inexpensive way by noting that for each *i* ($i \in I$), the index *q* associated to the variable z_i^q need not be less than the value w_i . This decreases the range of variation of the index q in this variable and the model can be rewritten as follows:

$$(DMR)\ \text{Min} \quad f_1 \sum_{i:w_i \le b_1} \sum_{q=w_i}^{b_1} z_i^q + \sum_{k=2}^{m} f_k \sum_{i:w_i \le b_k} \sum_{q=\max\{b_{k-1}+1, w_i\}}^{b_k} z_i^q \qquad (20)$$

$$\text{s. to} \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad j \in J \qquad (2)$$

$$\sum_{q=w_i}^{b_m} z_i^q \le 1 \qquad i \in I \qquad (21)$$

$$\sum_{j=1}^{n} w_j x_{ij} = \sum_{q=w_i}^{b_m} q z_i^q \qquad i \in I \qquad (22)$$

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

$$z_i^q \in \{0,1\} \qquad i \in I,\ q \in \{w_i,\ldots,b_m\} \qquad (23)$$

Clearly, we have that

Result 3: $v(\overline{DM}) \le v(\overline{DMR})$

The following example shows that the inequality stated in the previous result can be strict for some instances. Consider an instance for VSBPP with 3 items weighting 1, 2 and 3. Suppose that bins with capacities 1, 2 and 3 are available with a cost (per bin) of 1, 3 and 4, respectively for the mentioned capacities. The optimal value of this instance has cost equal to 8 and considers 2 bins of size 3 with the first two items packed in one of them and the third packed in the other. The optimal solution to $\overline{DM}$ has cost equal to 7.5 and is given by:

$$z_1^3 = 1,\ z_2^1 = 0.5,\ z_2^3 = 0.5,\ z_3^1 = 1$$

$$x_{12} = 1, x_{13} = 1/3, x_{23} = 2/3, x_{31} = 1$$

The optimal solution to $\overline{DMR}$ has value 23/3>7.5 and is given by:

$$z_1^1 = 1,\ z_2^3 = 1,\ z_3^3 = 2/3$$

$$x_{12} = 0.5,\ x_{22} = 0.5,\ x_{23} = 2/3,\ x_{31} = 1,\ x_{33} = 1/3$$

As we can see, variables z_2^1 and z_3^1 (that are not included in $\overline{DMR}$) have a non zero value in the optimal solution of $\overline{DM}$.

3. 5. Valid Inequalities

We note first, that by using (19), the inequalities (12) and (13) given in Section 3.1 for model *P3* can be rewritten in terms of the model *DM* as follows:

$$\sum_{i=1}^{n} \sum_{q=w_i}^{b_m} z_i^q \geq \left\lceil \frac{\sum_{j=1}^{n} w_j}{b_m} \right\rceil \qquad (24)$$

$$x_{ij} \leq \sum_{q=\max\{w_i, w_j\}}^{b_m} z_i^q \qquad i \in I,\ j \in J \qquad (25)$$

It should be noted that as suggested by Martello and Toth (1990) for BPP, the right-hand side of (24) can be improved. The same can be done for VSBPP if we consider the capacity of the largest bin. This modified constraint (24) will be used in our computations.

Following the same type of reasoning as in Gouveia and Saldanha-da-Gama (2006), we introduce next several classes of valid inequalities for VSBPP. Let us start by considering equalities (22). Dividing each term by an integer value *p* (*p>1*) we obtain the following equivalent equalities:

$$\sum_{j=1}^{n} \frac{w_j}{p} x_{ij} = \sum_{q=w_i}^{b_m} \frac{q}{p} z_i^q \qquad i \in I,\ p = 2,\dots,b_m \qquad (26)$$

By rounding down each coefficient in the left-hand side term of (26) and, subsequently, by rounding down the right-hand side term (this can be done due (21) and (23)) we obtain:

$$\sum_{j=1}^{n} \left\lfloor \frac{w_j}{p} \right\rfloor x_{ij} \leq \sum_{q=w_i}^{b_m} \left\lfloor \frac{q}{p} \right\rfloor z_i^q \qquad i \in I \qquad (27)$$

By adding inequalities (27) for all $i \in I$ and taking (2) into account, we obtain:

$$\sum_{i=1}^{n} \sum_{q=\max\{w_i,p\}}^{b_m} \left\lfloor \frac{q}{p} \right\rfloor z_i^q \geq \sum_{j=1}^{n} \left\lfloor \frac{w_j}{p} \right\rfloor \qquad p = 2,\dots,w_n \qquad (28)$$

Using a similar reasoning, but starting by rounding up the coefficients on the left-hand side term in (26), then rounding up the right-hand side term and finally summing in $i \in I$ we obtain

$$\sum_{i=1}^{n} \sum_{q=w_i}^{b_m} \left\lceil \frac{q}{p} \right\rceil z_i^q \leq \sum_{j=1}^{n} \left\lceil \frac{w_j}{p} \right\rceil \qquad p = 2,\dots,b_m \qquad (29)$$

Thus, we have jus proved the following result.

Result 4: Inequalities (28) and (29) are valid for the VSBPP.

A related set of inequalities can also be obtained by considering (22) and (2). By, first, adding constraints (22) for all $i \in I$ and then using (2) we obtain

$$\sum_{j=1}^{n} w_j = \sum_{i=1}^{n} \sum_{q=w_i}^{b_m} q z_i^q \qquad (30)$$

Dividing by p (p>1) each term and adequately rounding down the coefficients as before we obtain

$$\sum_{i=1}^{n} \sum_{q=\max\{w_i,p\}}^{b_m} \left\lfloor \frac{q}{p} \right\rfloor z_i^q \leq \left\lfloor \frac{\sum_{j=1}^{n} w_j}{p} \right\rfloor \qquad p = 2,\ldots,b_m \qquad (31)$$

Similarly, by rounding up the coefficients we obtain one more set of inequalities:

$$\sum_{i=1}^{n} \sum_{q=w_i}^{b_m} \left\lceil \frac{q}{p} \right\rceil z_i^q \geq \left\lceil \frac{\sum_{j=1}^{n} w_j}{p} \right\rceil \qquad p = 2,\ldots,b_m \qquad (32)$$

and we have just proved result 4.

Result 5: Inequalities (31) and (32) are valid for the VSBPP.

It should be noted that when $p = b_m$, inequality (32) is exactly inequality (24). A careful examination shows that some inequalities (32) are dominated by others as we obtain many such inequalities with the same right-hand side. Thus, we only need to consider the inequality with the "stronger" left-hand side. A similar situation arises with respect to the other three sets of inequalities.

4. Computational Experience

In this section we present some computational experiments to assess the efficiency of the models proposed in the previous sections. We start by describing the instances we worked with and then we analyze the results obtained. For obtaining the optimal integer value of the given instances, we use the branch-and-bound procedure provided by the commercial package ILOG CPLEX 8.0 (2002). We have tested several formulations for VSBPP, namely *P3*, *DMR* and enhancements of them using the valid inequalities presented in the last subsection of the previous section. For each model, we give the CPU time for solving it to optimality (a maximum time limit of five hours was considered) and for solving the corresponding linear relaxation. We have also examined the gap provided by the linear programming relaxation of the models here proposed. All the tests were performed a PC with a Pentium IV processor, 2.6 GHz and 512 Mb of RAM.

4. 1. Test Instances

Because no benchmark instances were found in the literature for the VSBPP, two classes of instances were generated. With respect to the first class, the instances were generated according with the following rules:

- The number of items was chosen in the set $\{10,20,60\}$.
- The weights of the items were randomly generated according to a (discrete) uniform distribution in the set $\{1,2,\ldots,20\}$.
- The cost associated with each bin was set equal to $\lfloor 100\sqrt{b} \rfloor$, where b is the corresponding capacity, in order to reflect an economy of scale.
- For each combination obtained with the parameters above two sets of available capacities were considered: $\{20,30,40\}$ and $\{15,20,25,30,35,40\}$, which means, m=3 and m=6, respectively.
- For each combination of all parameters 5 instances were generated.

For the second class of instances, we have tried to bring to the variable size situation, the structure of the instances that can be found in the literature, for the simpler BPP. Accordingly, we considered 3 sets of instances available in the OR Library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html) namely U120, T60 and T120. The instances in U120 consist of 120 items of sizes uniformly distributed in (20,100) to be packed into bins of size 150. The instances in the sets T60 and T120 consist of 60 and 120 items, respectively, of sizes uniformly distributed in (25,50) to be packed into bins of size 100. In each of these sets, we picked the first 5 instances and divided by 10 the capacity of the bins and the dimension of the items. The values were then rounded down so that integer values were obtained. We have then considered two cases: i) with only one type of bin (the classical problem) and ii) three available bin capacities. In the latter situation we have considered capacities equal to 10, 15 and 20 for the U instances and 5, 10 and 15 for the T instances. The costs associated with the bins were generated similarly as for the instances in class 1.

4. 2. Evaluating the Results

Tables 1 – 6 describe the results obtained. Each table is associated with one specific value for *n*. Accordingly, tables 1, 2 and 3 present the results for the instances in the first class and for n=10, n=20 and n=60, respectively. Tables 4, 5 and 6 present the results for the instances in the second class namely for the instances obtained from U120, T60 and T120, respectively.

In the tables, the first column indicates the model to which the corresponding line refers to. The remaining columns are divided into two sets of three columns. Each one of these set is associated with a specific value for *m* (3 or 6). The corresponding values for the capacities available for bins have been described above. The column entitled *o. m. / t. l.* presents the number of instances (out of five) for which an out of memory error occurred when using the branch-and-bound procedure of ILOG CPLEX 8.0 (2002). The other value described in the tables, is the number of instances whose optimal solution was not found within the time limit. The column *CPU* presents the average of the CPU time required for obtaining the optimal solution (excluding the instances for which either an out of memory error occurred or no optimal solution was found within the time limit). Column *GAP* presents the gap, in percentage, of the bound provided by the linear relaxation.

We have not presented the CPU time required by the linear programming relaxation because apart from three cases, the corresponding CPU time was always less than 1 second. The three exceptions refer to table 6 namely to the models P3+(12)+(13) (3.7 seconds), DMR (2.6 seconds) and DMR+(25)+ (28)+(29)+(31)+(32) (4.6 seconds). In any case, the CPU time corresponding to solving the linear programming relaxation is not significant.

Concerning the bound provided by the linear programming relaxation, the discretized formulation enhanced with valid inequalities presented in section 3.5 produced 0 gaps for all the instances in class 1 with m = 3 and for all instances in class 2. Concerning the instances in class 1 with m = 6, the gap obtained with the enhanced *DMR* formulation is very close to 0. These results give a strong indication of what to do when fast and good lower bounds are required, for instance, to evaluate a feasible solution obtained by some heuristic method. Another important outcome of these results is that inequalities (13) and (25) are not worth considering as enhancements for *P3* and *DMR* (in terms of the bound provided by the linear relaxation).

Concerning obtaining the optimal integer solutions, we were able to obtain all of them but only by using *DMR* with valid inequalities (again, this is a strong indication that discretized models together with valid "discretized" inequalities are worth trying). There are two exceptions that occur when we have consider the inclusion of inequalities (25): the first can be seen in table 4, where *DMR*+(25)+(28)+(29)+(31)+(32) exceeded the time limit. The second exception refers to table 6 where the same model lead to one "memory error" and twice the time limit was exceeded. Nevertheless, this gives some evidence that constrains (25) may not help much in solving the problem optimally.

In all instances with m = 3 or m = 6, even when the models *P3*+(12) or *P3*+(12)+(13) were able to solve the instances optimally, the CPU required for optimality was greater than the CPU required by *DMR* (with valid inequalities). Another important aspect regards *DMR* alone. With m = 3 and m = 6 the corresponding results are even worse. For m=1, the worst results are produced with *P3*+(12)+(13).

Model	m=3			m=6		
	o.m. / t. l.	CPU (sec.)	Gap (%)	o.m. / t. l.	CPU (sec.)	Gap (%)
P3	0 / 0	11	7,6	0 / 0	668	7,2
P3+12	0 / 0	4	2,0	0 / 0	652	2,0
P3+12+13	0 / 0	15	2,0	0 / 0	560	2,0
DMR	2 / 0	149	7,6	1 / 0	1137	7,2
DMR +28+29+31+32	0 / 0	1	0	0 / 0	233	0,3
DMR +25+28+29+31+32	0 / 0	1	0	0 / 0	104	0,3

Table 1: Instances in class 1 – 10 items.

Model	m=3			m=6		
	o.m. / t. l.	CPU (sec.)	Gap (%)	o.m. / t. l.	CPU (sec.)	Gap (%)
P3	2 / 1	963	5,4	1 / 2	2782	4,7
P3+12	1 / 0	1139	1,1	1 / 2	190	0,7
P3+12+13	1 / 1	2158	1,1	1 / 1	1428	0,7
DMR	5 / 0	-	5,4	5 / 0	-	4,7
DMR +28+29+31+32	0 / 0	5	0	0 / 0	3	0,1
DMR +25+28+29+31+32	0 / 0	4	0	0 / 0	2	0,1

Table 2: Instances in class 1 – 20 items.

Model	m=3			m=6		
	o.m. / t. l.	CPU (sec.)	Gap (%)	o.m. / t. l.	CPU (sec.)	Gap (%)
P3	5 / 0	-	1,9	5 / 0	-	1,5
P3+12	3 / 1	20	0,5	3 / 1	3	0,2
P3+12+13	0 / 4	180	0,5	0 / 3	231	0,2
DMR	5 / 0	-	1,9	5 / 0	-	1,5
DMR +28+29+31+32	0 / 0	26	0	0 / 0	35	0,0
DMR +25+28+29+31+32	0 / 0	510	0	0 / 0	274	0,0

Table 3: Instances in class 1 – 60 items

Model	m=1			m=3		
	o.m. / t. l.	CPU (sec.)	Gap (%)	o.m. / t. l.	CPU (sec.)	Gap (%)
P3	0/0	20	1,1	4/1	-	0,8
P3+12	0/0	21	0	4/1	-	0,2
P3+12+13	0/0	1275	0	0/4	8915	0,2
DMR	0/0	45	1,1	4/1	-	0,8
DMR +28+29+31+32	0/0	112	0	0/0	246	0
DMR +25+28+29+31+32	0/0	717	0	0/1	5310	0

Table 4: Instances in class 2 (obtained from U120) – 120 items.

Model	m=1			m=3		
	o.m. / t. l.	CPU (sec.)	Gap (%)	o.m. / t. l.	CPU (sec.)	Gap (%)
P3	0/0	1	2,7	4/1	-	1,8
P3+12	0/0	2	0	3/1	790	0,6
P3+12+13	0/0	35	0	3/1	705	0,6
DMR	0/0	3	2,7	4/1	-	1,8
DMR +28+29+31+32	0/0	2	0	0/0	17	0
DMR +25+28+29+31+32	0/0	25	0	0/0	258	0

Table 5: Instances in class 2 (obtained from T60) – 60 items.

Model	m=1 o.m. / t. l.	m=1 CPU (sec.)	m=1 Gap (%)	m=3 o.m. / t. l.	m=3 CPU (sec.)	m=3 Gap (%)
P3	0/0	56	1,9	4/1	-	1,6
P3+12	0/0	113	0	4/1	-	0,5
P3+12+13	0/0	993	0	0/5	-	0,5
DMR	0/0	797	1,9	4/1	-	1,6
DMR +28+29+31+32	0/0	34	0	0/0	284	0
DMR +25+28+29+31+32	0/0	564	0	1/2	1716	0

Table 6: Instances in class 2 (obtained from T120) – 120 items.

5. Conclusion

In this paper we have proposed several formulations for the VSBPP. In particular, we have proposed a so-called discretized model that together with valid inequalities (based on the discretized variables) is able to solve many instances that are not solved when non-discretized models are considered.

References

Cornuejols G., R. Sridharan, and J. M. Thizy (1991), "A comparison of heuristics and relaxations for the capacitated plant location problem", European Journal of Operational Research 50, 280-297;

Friesen D. K. and M. A. Langston (1986), "Variable sized bin packing", SIAM Journal on Computing 15, 222-230;

Garey M. and D. Johnson (1979), "Computers and Intractability: A guide to the theory of NP-Completeness", W. H. Freeman, San Francisco;

Gouveia, L., (1995), "A 2n constraint formulation for the capacitated minimal spanning tree problem." Operations Research 43, 130-141;

Gouveia L. and F. Saldanha-da-Gama (2006), "On the capacitated concentrator location problem: a reformulation by discretization", Computers and Operations Research 33, 1242-1258;

ILOG CPLEX 8.0 (2002), "Using the Cplex Callable Library", ILOG Inc., Incline Village, Nevada;

Kang J. and S. Park (2003), "Algorithms for the variable sized bin packing problem", European Journal of Operational Research 147, 365-372;

Martello S. and P. Toth (1990), "Knapsack problems. Algorithms and computer implementations", Wiley;