An Exact Algorithm for the Two-Dimensional Strip Packing Problem

<table>
<thead>
<tr>
<th>Journal:</th>
<th><em>Operations Research</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID:</td>
<td>OPRE-2007-11-565</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Article</td>
</tr>
<tr>
<td>Date Submitted by the Author:</td>
<td>27-Nov-2007</td>
</tr>
</tbody>
</table>
| Complete List of Authors: | Boschetti, Marco; University of Bologna, Department of Mathematics  
                                  Montaletti, Lorenza; University of Bologna, Department of Mathematics |
| Keywords:      | Cutting stock/trim < Production/scheduling, Integer . <  
                                  Programming, Branch-and-bound < Algorithms < Integer . <  
                                  Programming                                      |
An Exact Algorithm for the Two-Dimensional Strip Packing Problem

Marco Antonio Boschetti
Department of Mathematics, University of Bologna, Via Sacchi 3, 47023 Cesena, Italy, marco.boschetti@unibo.it,

Lorenza Montaletti
Department of Mathematics, University of Bologna, Via Sacchi 3, 47023 Cesena, Italy, montaletti.l@libero.it,

The Two-Dimensional Strip Packing Problem (2SP) appears in many industries (like steel and paper industries) and consists of cutting a rectangular master surface, called strip, with a given width and infinite height, into a number of rectangular items, each with a given size. The items must be cut with their edges always parallel or orthogonal to the edges of the master surface (orthogonal cuts) and we assume that items have a fixed orientation. The objective is to cut all the given items minimizing the total height of the used strip.

In this paper we propose reduction procedures, lower and upper bounds and an exact algorithm for the 2SP. The new lower bounds are both combinatorial bounds and derived from different relaxations of a mathematical formulation of the 2SP. While, the new upper bounds are constructive heuristics based on different strategies to place the items into the strip. The new exact method is based on a branch and bound approach.

Computational results on different sets of test problems derived from the literature show the effectiveness of the new lower and upper bounds and of the new exact algorithm.

Subject classifications: Production/scheduling; cutting stock/trim; programming: integer; algorithms: branch and bound.

Area of review: Optimization

History: Sent November 2007

1. Introduction

The Two-Dimensional Strip Packing Problem (2SP) consists of cutting a rectangular master surface, called strip, with a given width W and infinite height, into n rectangular items, each having width w_j and height h_j. The items must be cut with their edges always parallel or orthogonal to the edges of the master surface (orthogonal cuts) and we assume that items have a fixed orientation.

No other restrictions are required such as guillotine cuts. The objective is to cut all the given items minimizing the total height of the used strip.

The 2SP finds many practical applications and arises when the master surface can be considered of infinite height. For example, 2SP appears in steel and paper industries where rolls of material are cut into small rectangles or in transportation where the length of the truck bed used is to minimize.

Martello et al. (2003) shows that an instance of One-Dimensional Bin Packing Problem (1BP), where n items of size w_j have to be allocated in the minimum number of bin of capacity W, can be
transformed into a 2SP instance by setting the item heights $h_j = 1$. Therefore, the 2SP is strongly NP-hard as well as the 1BP (see Garey and Johnson (1979)).

According to the typology introduced by Wäscher et al. (2006) the 2SP can be classified as \textit{Two-Dimensional Open Dimension Problem} (2-ODP).

In literature many heuristic approaches have been presented so far to solve the 2SP. The first greedy heuristics were based on \textit{Bottom-Left} (BL) and \textit{Bottom-Left-Fill} (BLF) approaches and were proposed by Baker et al. (1980) and by Chazelle (1983). Given an ordered list of items, BL and BLF approaches place each item in turn into the strip. Usually, BL heuristics put each item in the top-right corner and move it as far down and left as possible. While, BLF heuristics generate a list of positions from the emerging solution and place each item in the lowest and leftmost feasible position contained in the list and left-justify it. Hence, BLF heuristics are able to fill some empty areas in the strip that BL heuristics cannot fill, but BLF heuristics are computationally more expensive. However, for the interested reader, a detailed description of BL and BLF approaches can be found in Hopper and Turton (2001) or Burke et al. (2004).

Recently, a number of metaheuristics have been proposed for the 2SP, that make use of BL and BLF approaches as local search, while at each iteration the order of the items is updated following different frameworks. Genetic algorithms have been proposed by Jakobs (1996), Gómez and De la Fuente (2000), Liu and Teng (1999) and Yeung and Tang (2004), while Iori et al. (2003) present a genetic algorithm, a tabu search and a hybrid algorithm. Lesh et al. (2005) and Lesh and Mitzenmacher (2006) update the order of the items for their BL local search randomly according to a given probability distribution or using a \textit{BubbleSearch} approach, respectively.

Two completely different algorithms are proposed by Martello et al. (2003) and are called \textit{BUILD} and \textit{JOIN}. The algorithm \textit{BUILD} starts from the solution provided by a new relaxation of the 2SP, called \textit{One-dimensional Contiguous Bin-Packing Problem} (1CBP), and tries to recover a feasible layout. JOIN applies a BL heuristic and three level-oriented packing algorithms to a modified instance obtained from the original by joining pairs of items differing in height by no more than a given threshold.

Burke et al. (2004) propose a greedy heuristic based on a \textit{Best-Fit} (BF) approach. At each iteration the BF heuristic adds to the emerging solution the item that best fits the lowest available space within the strip. While, Burke et al. (2006) improve the BF heuristic making use of BLF and of metaheuristics. In particular, their improved algorithm applies the BF heuristic until $n - m$ items are packed, where $m$ is a parameter, then applies for the remaining $m$ items the BLF heuristic in a metaheuristic framework that generates different item orderings for the BLF. The metaheuristics
used are tabu search, simulated annealing and genetic algorithms. Burke et al. (2004) and Burke et al. (2006) allow rotations of 90°.

Bortfeldt (2006) presents a genetic algorithm where the solutions are not encoded, but the corresponding layouts of the items into the strip are fully defined and directly manipulated by means of specific genetic operators. His genetic algorithm can take into account of the orientation and of the guillotine constraints.

A Greedy Randomized Adaptive Search (GRASP) algorithm is proposed by Alvarez-Valdes et al. (2006). Similarly to the BF, at each iteration, their algorithm adds to the emerging solution the item that best fits the available spaces, but avoiding the rigidity of static list by randomizing the selection process. At the end an improving phase tries to correct some wrong decisions inherent to the randomized construction process. The algorithm of Alvarez-Valdes et al. (2006) does not allow rotations.

Belov et al. (2006) adapt heuristics for one dimensional packing problems to the 2SP and propose two iterative heuristics. The first one is based on a single-pass heuristic which fills every most bottom-left free space in a greedy fashion by solving a one-dimensional knapsack problem considering only item widths and assigning suitable pseudo-profits to the items using the Sequential Value Correction (SVC) method already used by Mukhacheva et al. (2000) and Belov and Scheithauer (2007). The second heuristic is based on the randomized framework BubbleSearch of Lesh and Mitzenmacher (2006). It generates different item sequences and applies a Bottom-Left-Right (BLR) algorithm, that is a simple modification of the BL heuristic.

Lower bounds for the 2SP have been proposed and experimentally evaluated by Martello et al. (2003), Belov et al. (2006) and Alvarez-Valdes et al. (2007). While, at our knowledge, only two exact methods have been proposed in the literature by Martello et al. (2003) and by Alvarez-Valdes et al. (2007).

Martello et al. (2003) describe simple lower bounds derived from geometric considerations and adapting results presented in the literature for the 1BP and a lower bound based on the 1CBP relaxation of the problem. The lower bounds of Martello et al. (2003) are also used by Iori et al. (2003) and Bortfeldt (2006).

Belov et al. (2006) propose a lower bound based on the Bar Relaxation of Scheithauer (1999), where the basic idea is to generate a set of one-dimensional packing patterns (bar patterns) in horizontal and vertical directions and then to model the problem by means of the generated bar patterns. Belov et al. (2006) compute both the horizontal and vertical bar relaxations and the overall lower bound is given by the maximum of the two values obtained.
The lower bounds proposed by Alvarez-Valdes et al. (2007) are based on geometric considerations and on relaxations of an integer formulation of the 2SP.

The exact method of Martello et al. (2003) is based on a branch and bound approach that makes use of the upper and lower bounds proposed in the same paper and is able to solve test problem instances from the literature involving up to 200 items.

Alvarez-Valdes et al. (2007) describe a branch and bound algorithm, where the structure of the search tree is the same of Martello et al. (2003), but the performance has been improved by using the aforementioned new lower bounds and new dominance conditions.

1.1 Contributions

In this paper we propose three reduction procedures, four lower bounds, three upper bounds and an exact method for the 2SP. Some reduction procedures, derived from other packing problems, try to update the size of the strip and of the items in order to improve some lower bounds. Another new procedure tries to fix some items in solution and to reduce the size of the instance. New lower bounds are both combinatorial bounds and derived from different relaxations of mathematical formulations of the 2SP. While, two new upper bounds are greedy heuristics based on different search strategies and a third new upper bound is a truncated branch and bound that makes use of a partial solution provided by one of the new lower bound. In order to further improve both lower and upper bounds two postprocessing are proposed. The new exact method is a branch and bound based on the reduction procedures and on the lower and upper bounds proposed in this paper.

In our view, the main contributions of this paper include:

(1) A new reduction procedure for the 2SP very effective as, for some instance sets, a large number of items are fixed in solution. For some instances it also provides the optimal solution.

(2) Three completely new lower bounds for the 2SP. One is a combinatorial bound based on considerations involving item heights. While, the other two lower bounds are based on relaxations of two different mathematical formulations of the 2SP. The latter two lower bounds are time consuming but they are very useful for the hardest instances.

(3) A new postprocessing to improve the lower bound that also makes use of the normal pattern principle (see section 2.1).

(4) Three new heuristic algorithms. One is an improvement of the Best-Fit approach proposed by Burke et al. (2004). While, the remaining two heuristics are completely new. Even a postprocessing for improving the upper bound is proposed.
(5) A branch and bound that follows the classical depth-search strategy as Martello et al. (2003) and Alvarez-Valdes et al. (2007), but using a different rule to generate nodes and a simple dynamic strategy to select the node to expand.

(6) Computational results on a large number of test instance sets that also include instances not considered in the previous papers, in particular for the exact algorithms proposed in the literature so far (e.g., instances gcut5–gcut13, where the strip width ranging between 500 and 3000). Furthermore, the new exact algorithm solve to optimality several instances not solved so far.

1.2 Outline

The remaining of this paper is organized as follows. In Section 2 we give the notation and the definitions used throughout the paper and some rules for reducing the problem complexity. New lower bounds and new heuristic algorithms are presented in sections 3 and 4, respectively, and the exact method is described in section 5. In section 6 we analyze the computational performance of the procedures proposed in this paper on test problems derived from the literature. Conclusions are given in the section 7.

2. Notation, Definitions and Reductions

A set of $n$ rectangular items of size $(w_i, h_i), i \in P = \{1, \ldots, n\}$, and a strip of width $W$ and infinite height are given. Each item $i$ has $w_i \leq W$ and a fixed orientation and must be packed with its edges always parallel or orthogonal to the edges of the strip without overlapping other items or the region outside the strip. We assume that the size of the items and the width of the strip are integers. The objective is to cut all the items of $P$ into the strip minimizing the overall height $H$ of the packing pattern.

The strip is located in the positive quadrant of the Cartesian coordinate system with its bottom left-hand corner placed in position $(0,0)$ and with its bottom and left-hand edges parallel to the $x$-axis and the $y$-axis, respectively. The position of each item $i$ is defined by the coordinates $(x_i, y_i)$ of its bottom left-hand corner, referred to as the origin of the item, into the Cartesian system.

2.1 Principle of Normal Patterns

The origin of an item $i \in P$ can be located in every integer point $(x_i, y_i)$ of the strip such that $0 \leq x_i \leq W - w_i$. However, this set of points $(x_i, y_i)$ can be reduced by applying the principle of normal patterns as used by Hertz (1972), who called them canonical dissections, and Christofides and Whitlock (1977).
The principle of normal patterns is based on the observation that, in a given feasible cutting pattern, any cut item can be moved to the left and/or downward as much as possible until its left-hand edge and its bottom edge are both adjacent to other cut items or to the edges of the strip. Let $X_i$ and $Y_i$ denote the subsets of all $x$-positions and $y$-positions, respectively, where piece $i$ can be positioned. Using the principle of normal patterns, the sets $X_i$ and $Y_i$ for each piece $i \in P$ can be computed as follows:

\[
X_i = \left\{ x = \sum_{k \in P \setminus \{i\}} w_k \xi_k : 0 \leq x \leq W - w_i, \ \xi_k \in \{0, 1\}, \ k \in P \setminus \{i\} \right\} \tag{1}
\]

and

\[
Y_i = \left\{ y = \sum_{k \in P \setminus \{i\}} h_k \xi_k : 0 \leq y \leq H - h_i, \ \xi_k \in \{0, 1\}, \ k \in P \setminus \{i\} \right\} \tag{2}
\]

where $H$ is a suitable upper bound to the optimal height of the strip.

A simple dynamic programming recursion for computing $X_i$ and $Y_i$, for every $i \in P$, is described in Christofides and Whitlock (1977).

2.2 Problem Reductions

In this paper we apply two types of reductions: we fix some items in solution and we modify the item and strip sizes. These reductions can improve the value of the lower bounds and help the heuristic and the exact algorithms in finding a good quality feasible solution.

2.2.1 Fix some items in solution

There are some instances where there is a subset of items $R \subseteq P$ that can be permanently allocated into the bottom of the strip before starting to solve the problem without changing the optimal solution value. The subset $R$ contains all items that cannot be placed side by side with at least another item of $P$, i.e. $R = \{ j \in P : W - w_j < w_i, \forall i \in P \setminus \{j\} \}$.

All items belonging to the set $R$ can be allocated into a stack in the bottom of the strip occupying an height equal to $H_R = \sum_{j \in R} h_j$.

The exact algorithm can solve the problem using the remaining items, $P = P \setminus R$. To recover the optimal solution of the original instance it is sufficient to join the solution of the reduced instance, of value $H^*$, with the stack of the items in $R$ and setting the optimal strip height equal to $H^* = H^* + H_R$.

This reduction can be further improved by generalizing the idea proposed in Alvarez-Valdes et al. (2007). They consider the items of $R' = \{ j \in P : w_j > \frac{W}{2} \}$ in non-increasing order of width and observe that if all items belonging to set $P_j = \{ k \in P : w_j + w_k \leq W \}, \ j \in R'$, can be allocated into a rectangle of size $(W - w_j, h_j)$ then all items in the set $P_j \cup \{j\}$ can be fixed in the solution.
in the bottom of the strip (see Figure 1(a)). This idea can be extended by considering a subset \( R \subseteq R' \) instead of a single item \( j \in R' \). We allocated into a stack the items of \( R \) for non-increasing widths and in normal pattern positions (i.e., left justified). Then we check if all the items of set \( P_R = \{ k \in P : \exists j \in R \text{ s.t. } w_j + w_k \leq W \} \) can be allocated in the remaining area in the right side of the stack of height \( \sum_{j \in R} h_j \) (see Figure 1(b)). To check if the items in \( P_R \) can be allocated in the remaining area we use the heuristic algorithms presented in section 4. Obviously, before to run the heuristic algorithms we use some simple lower bounds, described in section 3, to check in advance if a feasible packing cannot exist. Figure 1(b) reports an example where no items can be fixed in solution if only one item of \( R' \) is considered at a time.

This new reduction procedure also generalizes the one proposed by Clautiaux et al. (2007c) for the two-dimensional orthogonal packing problem.

2.2.2 Modify the strip width If a combination of items that fill exactly the strip width \( W \) does not exist, then there is a useless space that can be removed from the strip without modifying the optimal solution value. The strip width can be updated by solving the following subset sum problem \( SSP_W \):

\[
W_P = \max \left\{ \sum_{i \in P} w_i \xi_i : \sum_{i \in P} w_i \xi_i \leq W, \xi_i \in \{0, 1\}, i \in P \right\}
\]

(3)

\( SSP_W \) is solved using the algorithm proposed in Pisinger (1995) and the new strip width is \( W_P \), i.e. \( W = W_P \). This reduction has been also proposed by Alvarez-Valdes et al. (2007).

2.2.3 Modify the item widths It is straightforward to observe that a packing layout containing item \( j \) remains feasible if the width of \( j \) is increased as \( w_j = w_j + (W - W^*_j) \), where \( W^*_j \)
is the optimal solution cost of the following subset sum problem $SSP_w(j)$:

$$W_j = \max \left\{ w = \sum_{k \in P \setminus \{j\}} w_k \xi_k + w_j : w \leq W, \xi_k \in \{0, 1\}, k \in P \setminus \{j\} \right\}$$ (4)

Since we would like to maximize the number of updated widths, heuristically we consider the items ordered for non increasing width, i.e. $w_1 \geq w_2 \geq \ldots \geq w_n$.

More details on this reduction procedure can be found in Boschetti et al. (2002) and Boschetti and Mingozzi (2003a).

3. Lower bounds

In this section we present some lower bounds for the 2SP. First, we propose some extensions of well-know lower bounds, then we introduce new approaches derived from other similar packing problems. We end this section proposing a postprocessing to further improve the lower bound.

3.1 Simple lower bounds

Two simple lower bounds are proposed by Martello et al. (2003). The first lower bound, called continuous lower bound, is computed in linear time by considering the total area of the items to allocate into the strip of width $W$ as follows:

$$L_c = \left\lceil \sum_{i \in P} \frac{w_i h_i}{W} \right\rceil$$ (5)

While, another trivial lower bound is the height of the tallest item:

$$L_h = \max_{i \in P} \{ h_i \}$$ (6)

3.2 Lower bounds based on dual feasible functions

Originally, the concept of dual feasible function was introduced by Johnson (1973) and used for computing lower bounds for bin packing problems for the first time by Lueker (1983).

In the last years, dual feasible functions have been used to compute lower bounds for different packing problems by Fekete and Schepers (1997, 1998, 2000), Fekete et al. (2007), Boschetti and Mingozzi (2003a,b), Boschetti (2004), Carlier et al. (2007), Clautiaux et al. (2006, 2007b,c,d,e) and Alvarez-Valdes et al. (2007). Furthermore, dual feasible functions have been used to generate valid inequalities in Alves (2005) and Baldacci and Boschetti (2007). Recently, Clautiaux et al. (2007a) survey and analyze theoretically several dual feasible functions and also perform a computational comparison.
The basic idea is to transform the widths of the items and of the strip by a function \( f \), called *dual feasible function*, in such a way that if a set of items of the original instance can be packed side by side in the same vertical position, then they can be still packed even with the new widths. A dual feasible function can be defined as follows.

**Definition 1.** A function \( f : [0, W] \rightarrow [0, W'] \) is called dual feasible, if for any finite set \( S \) of nonnegative real numbers, we have: \( \sum_{w \in S} w \leq W \Rightarrow \sum_{w \in S} f(w) \leq f(W) = W' \).

In general the values of a dual feasible function do not depend by the instance data. However, there exist dual feasible functions, called *data-dependent dual feasible functions* by Carlier et al. (2007), where their values cannot be computed in advance but depend on the sizes of the items of the instance considered. A data-dependent dual feasible function can be defined as follows.

**Definition 2.** A function \( f : [0, W] \rightarrow [0, W'] \) is called data-dependent dual feasible, if for any subset of items \( S \subseteq P \), we have: \( \sum_{i \in S} w_i \leq W \Rightarrow \sum_{i \in S} f(w_i) \leq f(W) = W' \).

Given a dual feasible function \( f \) a valid lower bound for the 2SP can be obtained by solving the continuous lower bound as follows:

\[
L_f = \left\lfloor \frac{\sum_{i \in P} f(w_i)h_i}{f(W)} \right\rfloor \tag{7}
\]

The dual feasible functions used in this paper are described in the following.

**Dual Feasible Function 1.** Let \( \alpha \in \Omega_1 \subseteq \mathbb{N} \). A dual feasible function defined by Fekete and Schepers (1997, 1998, 2000) is:

\[
f^1_\alpha(w) = \begin{cases} 
  w, & \text{if } (\alpha + 1)\frac{w}{W} \in \mathbb{Z} \\
  \lceil (\alpha + 1)\frac{w}{W} \rceil \frac{W}{\alpha}, & \text{otherwise}
\end{cases} \tag{8}
\]

**Dual Feasible Function 2.** Let \( \alpha \in \Omega_2 = [1, \frac{W}{2}] \), \( \Omega_2 \subseteq \mathbb{N} \). A dual feasible function defined by Fekete and Schepers (1997, 1998, 2000), but indirectly used for the first time by Martello and Toth (1990) and proposed for the 2SP in Martello et al. (2003), is:

\[
f^2_\alpha(w) = \begin{cases} 
  W, & \text{if } w > W - \alpha \\
  w, & \text{if } \alpha \leq w \leq W - \alpha \\
  0, & \text{otherwise}
\end{cases} \tag{9}
\]

**Dual Feasible Function 3.** Let \( \alpha \in \Omega_3 = [1, \frac{W}{2}] \), \( \Omega_3 \subseteq \mathbb{N} \). A dual feasible function proposed by Carlier et al. (2007) and derived from the one presented in Boschetti (2004) for the three-dimensional bin packing problem is:

\[
f^3_\alpha(w) = \begin{cases} 
  2 \left( \left\lfloor \frac{W}{\alpha} \right\rfloor - \left\lfloor \frac{W - w}{\alpha} \right\rfloor \right), & \text{if } w > \frac{W}{2} \\
  \left\lfloor \frac{W}{\alpha} \right\rfloor, & \text{if } w = \frac{W}{2} \\
  \left\lfloor \frac{w}{\alpha} \right\rfloor, & \text{if } w < \frac{W}{2}
\end{cases} \tag{10}
\]
Dual Feasible Function 4. Let $\alpha \in \Omega_4 = [1, \frac{W}{2}]$, $\Omega_4 \subseteq \mathbb{N}$, and $I_\alpha = \{i \in P : w_i \geq \alpha\}$. A data-dependent dual feasible function proposed for the two-dimensional bin packing problem in Boschetti and Mingozzi (2003a) is:

$$f^4_\alpha(w) = \begin{cases} M(W, I_\alpha) - M(W - w, I_\alpha), & \text{if } \frac{W}{2} < w \leq W \\ 1, & \text{if } \alpha \leq w \leq \frac{W}{2} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where $M(w, I)$ is the maximum number of items of the set $I$ that can be packed side by side in a bin of width $w$. $M(w, I)$ can be computed by solving the following knapsack problem:

$$M(w, I) = \max \left\{ \sum_{k \in I} \xi_k : \sum_{k \in I} w_k \xi_k \leq w, \xi_k \in \{0, 1\}, k \in I \right\} \quad (12)$$

$M(w, I)$ can be solved in $O(n)$ if items of the set $I$ are sorted by non-decreasing width.

Other dual feasible functions have been proposed in Clautiaux et al. (2007a) and Alvarez-Valdes et al. (2007), but not used in this paper. Clautiaux et al. (2007a) derive these new dual feasible functions from the functions used in Burdett and Johnson (1977), Vanderbeck (2000) and Letchford and Lodi (2002) to compute valid inequalities. While, other data-dependent dual feasible functions have been proposed by Alvarez-Valdes et al. (2007).

Fekete and Schepers (1997, 1998, 2000) show that dual feasible functions can be also obtained by composing two or more dual feasible functions. Hence, a dual feasible function $f^{1\beta\delta}_{\alpha\beta}$ can be obtained as follows: $f^{1\beta\delta}_{\alpha\beta}(w) = f^{1\beta}(f^{\delta\beta}_{\alpha}(w))$.

In our computational results we compose the four dual feasible functions above described only with $f^2_\alpha$. Therefore, the overall lower bound is given by:

$$L^{BM}_{d\beta} = \max_{k=1,2,3,4} \max_{\beta \in \Omega_4} \left\{ \max_{\alpha \in \Omega_4} \left\{ \left[ \sum_{i \in P} f^{k,2\beta}_{\alpha\beta}(w_i) h_i \right] \right\} \right\} \quad (13)$$

Notice that the dual feasible function $f^2_\alpha$ for $\alpha = 1$ is an identity function, i.e. $f^1_\alpha(w) = w$. Furthermore, in our implementation we use $\Omega_1 = \{1, \ldots, W\}$ and $\Omega_k = \{w_i : w_i \leq \frac{W}{2}, i \in P\} \cup \{W - w_i : w_i > \frac{W}{2}, i \in P\}$, for $k = 2, 3, 4$.

3.3 Lower bounds based on item heights

In section 3.1 we have introduced a simple lower bound proposed by Martello et al. (2003) that is given by the height of the tallest item. This lower bound can be improved, but not dominated, by considering the item heights in a different way.

First we have to compute a lower bound on the number of layers required to pack all items. In the layer 1 there are items placed on the strip base (i.e., with a $y$-position equal to zero), and in
the layer \( k \) there are items that in every \( x \)-positions are placed on at most \( k - 1 \) items and in at least an \( x \)-position are placed on exactly \( k - 1 \) items.

To compute the lower bound on the number of layers, \( \text{Lay} \), we can use the continuous lower bound for the 1BP:

\[
\text{Lay} = \left\lceil \frac{\sum_{i \in L} w_i}{W} \right\rceil \tag{14}
\]

Since each layer has to be tall at least as the minimum item height, sorting items by non-decreasing height, a lower bound for the 2SP is equal to the sum of the first \( \text{Lay} \) minimum heights:

\[
L_h^1 = \sum_{k=1}^{\text{Lay}} h_k \tag{15}
\]

Lower bound \( L_h^1 \) can be further improved by the following considerations. If the first \( \text{Lay} - 1 \) layers are completely filled for the entire width \( W \), then the top layer, i.e. the layer \( \text{Lay} \), contains items covering a width equal to \( w_{\text{Top}} = (\sum_{i \in L} w_i) \% W \), where \( \% \) is the modulo operator that is the remainder of division. While, if the first \( \text{Lay} - 1 \) layers are not completely filled then the remaining layers contain items covering a width greater than \( w_{\text{Top}} \).

We define the set \( I_{\text{Full}} \subseteq P \) as the set of \( \text{Lay} - 1 \) items of minimum height, i.e. \(|I_{\text{Full}}| = \text{Lay} - 1\) and \( h_i \leq h_j \), for each \( i \in I_{\text{Full}} \) and \( j \in P \setminus I_{\text{Full}} \). While, we define the set \( I_{\text{Top}} \) of \( k \) items of minimum height of \( P \setminus I_{\text{Full}} \) such that \( \sum_{i \in I_{\text{Top}}} \geq w_{\text{Top}} \), but \( \sum_{i \in I_{\text{Top}} \setminus \{j\}} < w_{\text{Top}} \), where \( j \) is the tallest item of \( I_{\text{Top}} \). A valid lower bound for the 2SP can be computed as follows:

\[
L_h^2 = \sum_{k \in I_{\text{Full}}} h_k + \max_{k \in I_{\text{Top}}} \{h_k\} \tag{16}
\]

Similarly, we define the set \( I'_{\text{Top}} \subseteq P \) of \( k \) items of minimum height such that \( \sum_{i \in I'_{\text{Top}}} \geq w_{\text{Top}} \), but \( \sum_{i \in I'_{\text{Top}} \setminus \{j\}} < w_{\text{Top}} \), where \( j \) is the tallest item of \( I'_{\text{Top}} \). While, we define the set \( I'_{\text{Full}} \subseteq P \setminus I'_{\text{Top}} \) as the set of the remaining \( \text{Lay} - 1 \) items of minimum height, i.e. \(|I'_{\text{Full}}| = \text{Lay} - 1\) and \( h_i \leq h_j \), for each \( i \in I'_{\text{Full}} \) and \( j \in P \setminus (I'_{\text{Full}} \cup I'_{\text{Top}}) \). Another valid lower bound for the 2SP can be computed as follows:

\[
L_h^3 = \sum_{k \in I'_{\text{Full}}} h_k + \max_{k \in I'_{\text{Top}}} \{h_k\} \tag{17}
\]

There are not dominance relations between lower bounds \( L_h^2 \) and \( L_h^3 \), while both dominate \( L_h^1 \). Hence, the overall lower bounds for the 2SP is:

\[
L_h^{BM} = \max \{ L_h^1, L_h^2, L_h^3 \} \tag{18}
\]
## 3.4 Lower bounds based on mathematical formulations

In this section we propose two lower bounds based on mathematical formulations used by Baldacci and Boschetti (2007) and Boschetti et al. (2002) for the *two-dimensional non-guillotine cutting problem* (NGCP).

### 3.4.1 Lower bound $L_{BM}^{F1}$

Let $X$ be the set of all $x$-coordinates along the strip width (i.e., $X = \{x: 1 \leq x \leq W, x \text{ integer}\}$) and let $Y$ be the set of all $y$-coordinates along the strip height (i.e., $Y = \{y: 1 \leq y \leq U, y \text{ integer}\}$, where $U$ represents a valid upper bound to the optimal optimal strip height). Moreover, remind that $X_i$ and $Y_i$ are the subset of all $x$ and $y$-positions where item $i$ can be placed following the principle of normal patterns (see section 2.1).

Let $t_{ix}$ and $s_{iy}$ be (0-1) binary variables both equal to 1 if and only if the bottom-left corner of the item $i$ is placed in position $(x, y)$ and let $\theta_{ixy}$ be a (0-1) binary variable equal to 1 if and only if position $(x, y)$ of the strip is covered by the item $i$.

Adapting the mathematical formulation for solving the feasibility problem for the NGCP proposed by Baldacci and Boschetti (2007), we can obtain the following formulation:

\[
(F1) \quad z_{F1} = \min h \\
\text{s.t.} \quad \sum_{x \in X_i} t_{ix} = 1, \quad i \in P \tag{20} \\
\sum_{y \in Y_i} s_{iy} = 1, \quad i \in P \tag{21} \\
\sum_{y \in Y_i} ys_{iy} \leq h - h_i, \quad i \in P \tag{22} \\
\sum_{y \in Y} \theta_{ixy} - h_i \sum_{k=a(i,x)}^x t_{ik} = 0, \quad i \in P, x \in X \tag{23} \\
\sum_{x \in X} \theta_{ixy} - w_i \sum_{k=b(i,y)}^y s_{ik} = 0, \quad i \in P, y \in Y \tag{24} \\
\sum_{i \in P} \theta_{ixy} \leq 1, \quad x \in X, y \in Y \tag{25} \\
t_{ix}, s_{iy}, \theta_{ixy} \in \{0, 1\}, \quad i \in P, x \in X, y \in Y \tag{26} \\
h \geq 0 \tag{27}
\]

where $a(i, x) = \max[0, x - w_i + 1]$ and $b(i, y) = \max[0, y - h_i + 1]$.

Constraints (20) and (21) impose that every item is packed exactly once into the strip. Constraints (22) impose that the height of the strip $h$ is sufficient to contain every items allocated in the $y$-positions specified by variables $s_{iy}$. Constraints (23) ensure that if an item $i$ is allocated in position $(x', y')$ then for each $x$-position in the range $[x', x' + w_i - 1]$ item $i$ must cover exactly


\[ z_{F1'} = \min h \]

\[ s.t. \quad \sum_{x \in X_i} t_{ix} = 1, \quad i \in P \]  

\[ \sum_{y \in Y_i} s_{iy} = 1, \quad i \in P \]  

\[ \sum_{i \in P} h_i \sum_{x \in \alpha(i,x)} t_{ix} \leq h, \quad x \in X' \]  

\[ \sum_{i \in P} w_i \sum_{y \in \beta(i,y)} s_{iy} \leq W, \quad y \in Y' \]  

\[ t_{ix}, s_{iy} \in \{0, 1\}, \quad x \in X_i, y \in Y_i, i \in P \]  

\[ h \geq 0 \]  

where \( X' = \bigcup_{i \in P} X_i \) and \( Y' = \bigcup_{i \in P} Y_i \), while \( \alpha(i, x) = \{ \tilde{x} \in X' : a(i, x) \leq \tilde{x} \leq x \} \) and \( \beta(i, y) = \{ \tilde{y} \in Y' : b(i, y) \leq \tilde{y} \leq y \} \), in order to consider only normal pattern positions. Problem \( F1' \) can be further reduced by observing that variable \( h \) (hence, even the objective function) is independent from variables \( \{ s_{iy} \} \), therefore the formulation becomes:

\[ z_{F1''} = \min h \]

\[ s.t. \quad \sum_{x \in X_i} t_{ix} = 1, \quad i \in P \]  

\[ \sum_{i \in P} h_i \sum_{x \in \alpha(i,x)} t_{ix} \leq h, \quad x \in X' \]  

\[ t_{ix} \in \{0, 1\}, \quad x \in X_i, i \in P \]  

\[ h \geq 0 \]  

The optimal solution value of problem \( F1'' \) is a valid lower bound, denoted by \( L_{BM}^{F1} \), and it can be computed by a MIP solver as CPLEX. In order to improve the performance of the MIP solver avoiding equivalent solutions, we can add to \( F1'' \) the following constraints:

\[ \sum_{j \in P(i,x)} t_{j,x-w_j} \geq t_{ix}, \quad x \in X_i, i \in P \]
where \( P(i, x) = \{ j \in P \setminus \{ i \} : x - w_j \in X_j \} \). Constraints (40) ensure that only solutions satisfying the normal pattern principle are considered. In our computational results we have added constraints (40) only if they are less than \( \max\{500, \frac{1}{2}nW\} \).

Since the computing time available for solving problem \( F1'' \) cannot exceed a given time limit, if the optimal solution is not reached we set the lower bound \( L_{BM} \) equal to the minimum lower bound associated to the remaining unexplored nodes got from the MIP solver. In our computational results the time limit for the MIP solver is 600 seconds.

Starting from mathematical formulation \( F1 \) other relaxed problems can be obtained. In particular, using only the variable set \( \{ s_{iy} \} \) can be obtained a mathematical formulation equivalent to the 1CBP proposed by Martello et al. (2003), but our computational results, not reported in this paper, show that it is not competitive with respect to \( F1'' \).

### 3.4.2 Lower bound \( L_{F2}^{BM} \)

The relaxed problem proposed in this section is based on the observation that any feasible 2SP solution can be represented by a sequence \( (S_1, \ldots, S_h) \), where each \( S_y \in \mathcal{Y} \) is the subset of items covering the \( y \)-position \( y' \).

**Definition 3.** A subset of items \( S \subseteq P \) is a \( y \)-feasible subset if \( \sum_{i \in S} w_i \leq W \).

Let \( \mathcal{Y} = \{1, \ldots, m\} \) be the index set of all \( y \)-feasible subsets and let \( \mathcal{Y}_i \subseteq \mathcal{Y} \) be the index set of all \( y \)-feasible subsets containing item \( i \in P \). However, it is not necessary consider all \( y \)-feasible subsets, as shown by Boschetti et al. (2002) for the NGCP, but only undominated \( y \)-feasible subsets that are defined as follows.

**Definition 4.** A \( y \)-feasible subset \( S_j, j \in \mathcal{Y}, \) is undominated if \( S_j \not\subseteq S_k, \forall k \in \mathcal{Y} \setminus \{j\} \).

We denote with \( \bar{\mathcal{Y}} \) the index set of undominated \( y \)-feasible subsets, i.e. \( \bar{\mathcal{Y}} = \{ j \in \mathcal{Y} : S_j \not\subseteq S_k, \forall k \in \mathcal{Y} \setminus \{j\} \} \). Moreover, \( \bar{\mathcal{Y}}_i = \bar{\mathcal{Y}} \cap \mathcal{Y}_i \). Let \( v_j \) be the number of times that the undominated \( y \)-feasible subset \( j \in \bar{\mathcal{Y}} \) is in the optimal solution. A valid lower bound for the 2SP can be computed by solving the following problem:

\[
(F2) \quad z_{F2} = \min \sum_{j \in \bar{\mathcal{Y}}} v_j \tag{41}
\]

\[
\text{s.t.} \quad \sum_{j \in \mathcal{Y}_i} v_j \geq h_i, \quad i \in P \tag{42}
\]

\[
v_j \geq 0, \quad j \in \bar{\mathcal{Y}} \tag{43}
\]

Constraints (42) ensure that in the optimal solution every item \( i \) is contained in at least \( h_i \) undominated \( y \)-feasible subsets. While, the objective function (41) minimizes the number of undominated \( y \)-feasible subsets used.
Problem $F_2$ can be solved by an LP solver as CPLEX and its optimal solution value is the valid lower bound, denoted by $L_{BM}^{F_2}$. However, if the number of $y$-feasible subsets (i.e., $|\bar{Y}|$) is huge we can generate only a subset of $\bar{Y}$, but a column generation is required to add the missing $y$-feasible subsets used in the optimal $F_2$ solution. Since a column generation needs to solve a knapsack problem at each iteration and for an instance where $|\bar{Y}|$ is huge many iterations are required, if we are not able to generate the entire set $\bar{Y}$ we set $L_{BM}^{F_2} = 0$.

In our computational results $\bar{Y}$ cannot exceed one million of subsets and the time limit for the LP solver is 600 seconds.

3.5 A postprocessing to improve the lower bound

In this section we propose a simple postprocessing to improve the current best lower bound, denoted by $LB$.

The first improvement of $LB$ can be obtained by noticing that the required strip height must be a normal pattern position computed by considering all items $P$. Therefore, we can update the current lower bound as follows:

$$LB = \min \left\{ y = \sum_{k \in P} h_k \xi_k : LB \leq y \leq UB, \xi_k \in \{0,1\}, k \in P \right\}$$

(44)

where $UB$ is the current best upper bound that is certainly a normal pattern position.

A more expensive procedure to improve the lower bound $LB$ fixes the strip height at value $\hat{h} \in [LB, UB]$ and computes a lower bound on the strip width required to allocate all items $P$. To compute such a lower bound we rotate the instance exchanging widths with heights and we apply lower bounds $L_{BM}^{h}$, $L_{BM}^{w}$, and $L_{BM}^{F_2}$, described in the previous sections, to the resulting problem. We do not use $L_{BM}^{F_1}$, because it is expensive, and for $L_{BM}^{F_2}$ we set a time limit of only 60 seconds and we generate at most half a million undominated $y$-feasible subsets, i.e., $|\bar{Y}| \leq 500000$. If the lower bound on the strip width computed is less or equal to $W$ it means that all items can be packed in a strip of size $(W, \hat{h})$, otherwise height $\hat{h}$ is not sufficient to contain all items of the problem instance and $LB$ can be increased to the smaller normal pattern height greater than $LB$ by setting $LB = \hat{h} + 1$ and applying expression (44).

This search procedure starts setting $\hat{h} = UB - 1$ and, at each iteration, $\hat{h}$ is decreased by 1 unit. The procedure terminates only if items $P$ cannot be packed in a strip of size $(W, \hat{h})$, hence, in the worst case, this procedure is iterated $UB - LB$ times.
4. Heuristics

In this section we propose three new heuristic algorithms. The first one is based on original ideas and the principle of normal patterns, while the second one is a modified version of algorithms already proposed in the literature. The third heuristic algorithm makes use of the lower bound $L_{BM}^{BF1}$, described in section 3.4.1.

The first two heuristic procedures add at each iteration an item to the emerging solution $E$. At the beginning the emerging solution is set empty, i.e., $E = \emptyset$. When at each iteration an item $i$ is placed into the strip in position $(x_i, y_i)$, the procedures add to the emerging solution $E$ the triplet $(i, x_i, y_i)$, i.e., $E = E \cup \{(i, x_i, y_i)\}$.

The third procedure fixes the $x$-positions computed for all items by lower bound $L_{BM}^{BF1}$ and tries to find the $y$-positions by a truncated branch and bound.

At the end of this section we also propose a postprocessing to improve the current best upper bound.

4.1 Normal Pattern Shifting (NPS)

The Normal Pattern Shifting (NPS) heuristic places items in turn into the strip following a given order. In our computational results we have ordered items using different criteria (e.g., non-increasing value of the area, height, etc.) and we have updated the order of items as shown in section 4.1.1.

To place each item into the strip the NPS procedure starts from seed positions. The seed positions are defined by items already allocated into the strip and are of three different types: Starting, Top-Left and Bottom-Right. The starting seed position is only one and corresponds to position $(0, 0)$. While, top-left and bottom-right seed positions correspond to the positions of top-left corners and of bottom-right corners of the items already allocated into the strip, respectively (see Figure 2).
At the beginning the strip is empty (i.e., $E = \emptyset$) and only the starting seed position $(0,0)$ is available. This position is always a normal pattern position and every items can be placed here. When, at each iteration, a new item $i$ is placed into the strip in position $(x_i, y_i)$, its top-left and bottom-right corners are added to the list of the available seed positions, called $ListSPos$. Notice that at each iteration the seed positions to consider are $O(n)$.

The NPS heuristic at each iteration finds for the item $i$ considered its normal pattern positions starting from the seed positions contained in $ListSPos$. If the search starts from a bottom-right seed position the procedure tries to move down the item $i$ as much as possible, while if it starts from a top-left seed position the procedure tries to move to left the item $i$ as much as possible (see Figure 3). No other movements are required because if the position obtained is feasible but further movements are required to reach a normal pattern position, there exists at least another seed position that allows to reach the same normal pattern position (see Figure 3(b)). If for the item $i$ a feasible normal pattern position $(x_i, y_i)$ is found (i.e., the item is inside the strip, does not overlap other items already allocated and its position satisfies the normal pattern principle), the procedure evaluates its quality $q(i, x_i, y_i, E)$ by computing the density of the corresponding emerging solution as $q(i, x_i, y_i, E) = \frac{\sum_{(j, x_j, y_j) \in E} w_j h_j}{W \bar{h}}$, where $\bar{h} = \max\{y_j + h_j : (j, x_j, y_j) \in E\}$. The NPS heuristic chooses the lowest and leftmost position that maximizes the density of the emerging solution.

The NPS heuristic can be modified to consider at each iteration all items not already allocated and to choose the item and the position that maximizes the density of the emerging solution. If more items and/or positions maximize the density the procedure chooses the item with the lowest and leftmost position. However, our computational results, not reported in this paper, show that
not enough improvements of the upper bounds are achieved in spite of a considerable increasing of the computing time.

4.1.1 Normal Pattern Shifting with Pricing  Procedure NPS can be repeated with different item orders generated by using a framework derived from the one proposed by Boschetti and Mingozzi (2003b) for the Two-dimensional Bin Packing problem (2BP).

At the beginning for every item $i \in P$ we set an initial price $p_i$ equal to its area, height, etc., or a fixed value (e.g., $p_i = 100$). We execute NPS with items sort by non-increasing price value and let $H$ be the height of the strip used. We use the position $(x_i, y_i)$ of each item $i \in P$ in the solution to increase or decrease its price. If the item $i$ is placed in the bottom half of the strip (i.e., $y_i \leq H/2$), its price is decreased: $p_i = \alpha p_i$, where $\alpha < 1$. While, if the item $i$ is placed in the top half of the strip (i.e., $H/2 < y_i \leq H$), its price is increased: $p_i = \beta p_i$, where $\beta > 1$. Values $\alpha$ and $\beta$ are randomly generated every time are required as follows: $\alpha = 1 - r$ and $\beta = 1 + r$, where $r$ is a random real number in the range $[0, 1)$.

Procedure NPS with pricing requires several iterations to give some improvements, thus it is time consuming. Therefore, in our computational results, we have used the pricing only for small instances, i.e., $n \leq 20$.

4.2 Priority Best-Fit (PBF)

The Priority Best-Fit (PBF) heuristic procedure is based on the BF algorithm of Burke et al. (2004) and makes use of some improvements proposed by Alvarez-Valdes et al. (2006).

Given an emerging solution $E$, the PBF heuristic identifies the available positions using a skyline data structure (see Burke et al. (2004) for more details). In Figure 4 is shown an example where in the emerging solution $E$ seven items have been already allocated into the strip. Figure 4(a) shows the skyline generated by $E$, where the free dark areas below the skyline cannot be used any more. The available positions are the gaps of the skyline as shown in Figure 4(b).

Procedure PBF selects the lowest available gap, also called niche, and tries to place there the item not already allocated that best fits the available space. If there are more gaps at the lowest height we select the leftmost. The gaps at the left and at the right of the niche are called left gap and right gap, respectively. While, both gaps are called neighbor gaps. In the example of Figure 4 the lowest gap is $Gap_4$ and its neighbors are $Gap_3$ and $Gap_5$.

To identify the best fitting item PBF procedure applies different placement criteria that can be classified in two categories: hard selective and weakly selective. The hard selective criteria choose an item that exactly matches some characteristics and we use the following: $(h,1)$ the item has the
same width of the niche; (h.2) the item has the same height of the niche, i.e. its top edge reaches one of the neighbor gaps; (h.3) the item reaches the same height of the left gap or, if the niche touches the left edge of the strip, it’s the tallest; (h.4) the item with another one, even of the different height, fill the whole niche width. While, we use the following weakly selective criteria: (w.1) the item has the largest height; (w.2) the item has the largest width; (w.3) the item has the largest area; (w.4) the item with another one of the same height maximizes the filled niche; (w.5) the item with another one, even of the different height maximizes the filled niche; (w.6) the item maximize the density of the emerging solution. If no one of remaining items can be allocated, because the width of the lowest available gap is too small, the space in the niche can be considered waste. Therefore, the lowest gap is deleted and its width is added to the lowest neighbor gap.

At each execution of the PBF heuristic we use a different ordered subset of the criteria previously listed. Not all combinations of criteria are necessary even because a lot of them are meaningless. Table 1 reports the combinations of criteria used in the PBF heuristic specifying the order in which they are applied. Each row describes a different combination by reporting in the columns corresponding to the criteria included the priority assigned (priority 1 means it is the first criterion applied, etc.).

When a criterion has selected an item of width smaller than the niche width, we have to choose between the left or the right alignment of the item inside the niche. We have used the same procedure proposed by Alvarez-Valdes et al. (2006). Let \( S_L \) and \( S_R \) be the heights of the left and right gaps, respectively. Let \( h^* \) be the \( y \)-position of the niche and \( h' = h^* + h_i \). The alignment procedure works as follows. If the niche touches the left edge of the strip or \( S_L = h' \) the item is aligned on the left. Otherwise, if \( S_R = h' \) the item is aligned on the right. If no one of the previous

![Figure 4](image_url) An example of skyline data structure: the lowest gap (niche) is Gap_4.
conditions occurs and $S_i = S_r$, the item is placed in the position nearest at the strip edge, otherwise
the item is aligned to the tallest neighbor gap.

Unfortunately, tallest items can be placed into the strip only at last iterations because the
placement criteria work to build a smooth solution. Thus, to avoid bad packing we apply the
following procedure similar to the one proposed by Alvarez-Valdes et al. (2006). Let $i$ be the
item selected by the placement criteria and let $j$ be the tallest of remaining items for which the
selected niche is feasible. Let $A_E$ be an estimation of the empty area after placing the tallest item $j$ and let $A_M$ be the sum of the areas of remaining items. Given the height $H$ of the current
packing and the $y$-position $\bar{y}$ of the niche, $A_E$ is computed by subtracting from $W(H - \bar{y})$ the
area of the items in the emerging solution $E$ covering the area of the strip from $\bar{y}$ to $H$, i.e. $A_E = W(H - \bar{y}) - \sum_{(k,x_k,y_k) \in E} w_k \Delta h'_k(\bar{y})$, where $\Delta h'_k(\bar{y}) = \max\{\bar{y}, y_k + h_k\} - \max\{\bar{y}, y_k\}$. If $A_E > A_M$ we
place $j$ into the strip, otherwise we perform a further look-ahead estimation to verify if packing the
item $i$ into the niche implies that at the next iteration $A_E > A_M$. In this case we place the item $j$,
otherwise we place $i$.

4.2.1 PBF with a warm start In order to improve the quality of the packing generated
by PBF procedure, we try to apply a procedure similar to the SubKP algorithm proposed by Belov
et al. (2006). SubKP builds a 2SP feasible solution slice after slice filling the empty spaces solving
a one-dimensional knapsack problem.

In our PBF procedure we take advantage to select only the first $y$-feasible subset of items placed
in the bottom of the strip, then we apply the PBF procedure described in the previous section.
Therefore, we repeat PBF procedure for $k$ different starting $y$-feasible subsets that maximize the
width covered. In particular, in our computational results we have chosen the $k$ $y$-feasible subsets of

<table>
<thead>
<tr>
<th>Comb.</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h.1 h.2 h.3 h.4 w.1 w.2 w.3 w.4 w.5 w.6</td>
</tr>
<tr>
<td>C1</td>
<td>2 1 3</td>
</tr>
<tr>
<td>C2</td>
<td>1 2 3</td>
</tr>
<tr>
<td>C3</td>
<td>1 3 2</td>
</tr>
<tr>
<td>C4</td>
<td>1 3 2</td>
</tr>
<tr>
<td>C5</td>
<td>1 2 3</td>
</tr>
<tr>
<td>C6</td>
<td>1 2 3</td>
</tr>
<tr>
<td>C7</td>
<td>1 2 3</td>
</tr>
<tr>
<td>C8</td>
<td>2 3 1</td>
</tr>
<tr>
<td>C9</td>
<td>1 2</td>
</tr>
<tr>
<td>C10</td>
<td>2 1</td>
</tr>
<tr>
<td>C11</td>
<td>1 2</td>
</tr>
<tr>
<td>C12</td>
<td>1 2 3</td>
</tr>
<tr>
<td>C13</td>
<td>2 1</td>
</tr>
<tr>
<td>C14</td>
<td>1 2</td>
</tr>
<tr>
<td>C15</td>
<td>1 2 3</td>
</tr>
<tr>
<td>C16</td>
<td>1 3</td>
</tr>
<tr>
<td>C17</td>
<td>2 3</td>
</tr>
<tr>
<td>C18</td>
<td>3 1</td>
</tr>
<tr>
<td>C19</td>
<td>1 2</td>
</tr>
<tr>
<td>C20</td>
<td>2 3</td>
</tr>
</tbody>
</table>

Table 1 Combinations of criteria used at each execution of the PBF heuristic. The number reported in each row specifies the order in which they are applied.
width equal to \( W \) and \( k \) is limited to \( \left\lfloor \frac{MS}{n} \right\rfloor \). In our computational results we have fixed \( MS = 2 \times 10^6 \), in order to have \( k = 5000 \), for instances with \( n = 20 \), and \( k = 200 \), for \( n = 100 \).

We have also tried to iterate the procedure to complete the solution by adding a \( y \)-feasible subset at each iteration slice after slice, as in Belov et al. (2006), but the results are not improved in spite of the increased computing time. While, Belov et al. (2006) have obtained very good results combing the SubKP with the iterative scheme SVC.

### 4.3 Heuristic based on lower bound \( L^{BM}_{F1} \) (HF1)

The third procedure HF1 fixes the \( x \)-positions computed for all items by lower bound \( L^{BM}_{F1} \) and tries to find the \( y \)-positions by a truncated branch and bound.

Lower bound \( L^{BM}_{F1} \) is computed by solving problem \( F1'' \). Hence, if the MIP solver is not able to solve \( F1'' \) within the given time limit, in order to improve the heuristic solution of \( F1'' \) we apply the polishing procedure available in CPLEX setting a time limit of 60 seconds.

Given the best solution found solving \( F1'' \), hopefully the optimal, we fix the \( x \)-positions of the items according such a solution and we apply a truncated version of the branch and bound described in section 5 where only the \( y \)-positions or each item must be fixed. It is always possible to find a feasible solution if some requirements are relaxed such as normal pattern principle, etc. While, there is not proof that a feasible 2SP solution of height equal to \( L^{BM}_{F1} \) exist. Otherwise, \( L^{BM}_{F1} \) should be the optimal solution.

The truncated branch and bound with respect to the exact method described in section 5 makes use only of the continuous lower bound and when it generates the child nodes for each item it considers only the positions having the \( x \)-coordinate fixed by lower bound \( L^{BM}_{F1} \). Moreover, in our computational results, we set a small time limit of 60 seconds.

### 4.4 A postprocessing to improve the upper bound

The current best upper bound \( UB \) can be further improved applying a postprocessing described in this section.

Let \( LB \) be the best lower bound computed so far and let \( \hat{h} \in [LB, UB] \). We rotate the problem instance by 90° by exchanging for each item \( i \in P \) the width \( w_i \) and the height \( h_i \) and by setting the strip width equal to \( \hat{h} \). If there exists a heuristic solution where every items can be packed in a strip of height less than or equal to \( W \) then \( UB \) can be updated by setting \( UB = \hat{h} \).

We propose a sequential search in the range \( [LB, UB] \). The search procedure starts by setting \( \hat{h} = LB \) and, at each iteration, \( \hat{h} \) is increased by 1 unit. The procedure terminates as soon as items
5. Exact method

The new exact algorithm, called BB, is a branch and bound that makes use of reduction procedures, described in section 2.2, and of lower and upper bounds, described in sections 3 and 4, respectively.

Let LB and UB be the current best lower and upper bounds, respectively. We initialize LB = 0 and LB = +∞. Every time a new lower bound or a new upper bound is computed LB and UB are updated accordingly. As soon as LB = UB an optimal solution is found and the algorithm stops.

Algorithm BB, at the root node after reductions, applies the lower bounds LM and LB and then the heuristic procedures NPS and PBF. If the optimal solution is not found (i.e., LB < UB) algorithm BB computes first LM and then LB. If within the given time limit the MIP solver has found at least a feasible solution for F1, BB applies the heuristic procedure HF1. Algorithm BB ends the root node with the two postprocessing to improve the current best lower and upper bounds. If at the root node the optimal solution is not found, algorithm BB starts the branching.

At the beginning, no items have been placed yet and the strip is empty. Therefore, according to the normal pattern principle, items can be placed only in position (0, 0). Thus, BB generates exactly n child nodes, one for each item placed in (0, 0). While, at each node of the search tree, the remaining items can be placed only in particular positions, called corners. A corner is computed using the skyline data structure, described in section 4.2. Remind that the skyline is a set of consecutive segments positioned at different y-positions and each segment represents a gap. Therefore, each gap k is represented by the triplet (x_l^k, y_k), where x_l^k and x_r^k are the x-positions of the left and right ends of the gap, respectively, and y_k is its y-position. The left end x_l^k of the gap k generates a corner (x_l^k, y_k) if it is next to the left-hand edge of the strip (i.e., x_l^k = 0) or the previous gap has a greater y-position. The algorithm selects the corner that minimizes the y-position and, among corners with the same minimum y-position, the corner that minimize the x-position. Algorithm BB generates at most n + 1 child nodes, one for each remaining item placed in the selected corner and one if the selected corner is not used. However, algorithm BB generates a child node only if placing the item into the corner is feasible and the normal pattern principle is satisfied. Furthermore, if more items have the same size only one child node is generated.

For each child node generated, BB updates the skyline data structure and computes a lower bound on the remaining problem using LM, LB and LB, if Y does not exceed 200000 subsets and with a time limit of 1 second. The postprocessing is applied only if the current lower bound is
less than $W$. If the overall lower bound is greater or equal to the current best upper bound the
node is fathomed.

Algorithm BB follows a depth-first approach. When it has expanded a node, if no child nodes
have been generated BB backtracks, otherwise it has to select one of its child nodes to continue
the tree search. In our computational experiments we have observed that the order of the child
nodes is very important for the overall performance of BB. Therefore, we propose the following
dynamic strategy. If the 40% of the remaining items have a width larger than $\bar{w}$, we sort the child
nodes by non-increasing heights of the corresponding items. Otherwise, we sort the child nodes by
non-increasing widths. In our computational results we have set $\bar{w} = \frac{W - w_{\text{min}}}{2}$, where $w_{\text{min}}$ is the
smaller width of the remaining items.

6. Computational results

The algorithms presented in this paper have been coded in ANSI C using Microsoft Visual Studio 6
and run on a Laptop equipped with an Intel Pentium M 725 1.60 GHz. Ilog CPLEX 10.1 was used
as mixed integer linear programming solver for computing lower bound $L_{BM_1}^{BM}$, described in section
3.4.1, and as linear programming solver for computing lower bound $L_{BM_2}^{BM}$, described in section 3.4.2.

We have considered the following set of test problems: $nycut$, 12 instances from Beasley (1985b);
$cgcut$, 3 instances from Christofides and Whitlock (1977); $gcut$, 13 instances from Beasley (1985a);
$beng$, 10 instances from Bengtsson (1982); $ht$, 9 instances from Hopper and Turton (2001); $bkw$,
13 instances from Burke et al. (2004); $class$, 500 instances subdivided in 10 classes, where each
class contains 5 groups, that differ for the different number of items, and each group contains 10
instances. The first 6 classes were proposed by Berkey and Wang (1987), the other 4 classes were
proposed by Martello and Vigo (1998).

The two sets $ht$ and $bkw$ are obtained by cutting a starting master $(W, H)$ into smaller rectangles,
so the optimal solution value is always equal to continuous bound, i.e. equal to $H$. While, the
remaining sets are well known test instances for other two-dimensional packing problems that have
been transformed into strip-packing instances by setting the strip width $W$ equal to the width of
the master surface or of the bin.

Detailed tables are collected in the online supplement Boschetti and Montaletti (2007). While, in
Tables 2–5 we report in each row the overall results (i.e., averages or sums) of a set of test instances
whose name is reported in column $Name$. For test instances $class$ we report the results of each
class in a different row. When in our tables a cell is empty it means that the value is not available.
Table 2 Reduction test procedures, described in section 2.2.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Reductions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Δn%</td>
</tr>
<tr>
<td>ncut 10-22 10-30</td>
<td>7.19</td>
</tr>
<tr>
<td>cgcut 16-62 10-70</td>
<td>0.00</td>
</tr>
<tr>
<td>gcut 10-50 250-300</td>
<td>31.69</td>
</tr>
<tr>
<td>beng 20-200 25-40</td>
<td>0.00</td>
</tr>
<tr>
<td>ht 16-29 20-60</td>
<td>0.00</td>
</tr>
<tr>
<td>bkw 10-3152 40-640</td>
<td>8.19</td>
</tr>
<tr>
<td>Class 1 20-100 10</td>
<td>53.43</td>
</tr>
<tr>
<td>Class 2 20-100 30</td>
<td>0.00</td>
</tr>
<tr>
<td>Class 3 20-100 40</td>
<td>44.87</td>
</tr>
<tr>
<td>Class 4 20-100 100</td>
<td>0.00</td>
</tr>
<tr>
<td>Class 5 20-100 100</td>
<td>63.89</td>
</tr>
<tr>
<td>Class 6 20-100 300</td>
<td>0.00</td>
</tr>
<tr>
<td>Class 7 20-100 100</td>
<td>91.20</td>
</tr>
<tr>
<td>Class 8 20-100 100</td>
<td>99.17</td>
</tr>
<tr>
<td>Class 9 20-100 100</td>
<td>16.50</td>
</tr>
</tbody>
</table>

While, the character “_” is used when the corresponding value cannot be computed (e.g., if no instances have been solved the average computing time for the solved instances is not available).

6.1 Reductions

Table 2 shows the results of the reductions procedures and the details of the test instances reporting for each set of test instances the minimum and maximum number of items in column n and the minimum and maximum width of the strip in column W.

The reduction procedure that fix items in solution, described in section 2.2.1, can be evaluated considering the percentage reduction of the number of items Δn%, i.e., $\Delta n% = \frac{n - n'}{n} \times 100$, where n’ is the number of items to allocate after reductions, and the percentage of height of the strip containing items fixed in the solution ΔH%, i.e., $\Delta H% = \frac{H - H'}{H} \times 100$, where $H'$ is the best height found by the new exact algorithm. While, to evaluate the reductions that modify the width of the strip and of the items, described in sections 2.2.2 and 2.2.3, column ΔW% shows the percentage reduction of the strip, i.e., $\Delta W% = \frac{W - W'}{W} \times 100$, where $W'$ is the strip width after reductions, and column ΔA% shows the percentage increasing of the item areas, i.e., $\Delta A% = \frac{A_f - A'_f}{A_f} \times 100$, where $A_f$ and $A'_f$ are the total area of items before and after reductions, respectively.

The procedures that modify the strip and item sizes give a small contribution, while in several sets of test instances many items are fixed in solution and for some instances the optimal solution is reached. The computing time is negligible.

6.2 Lower bounds

Table 3 compares new lower bounds, described in section 3, and lower bounds proposed in the literature.

Columns $G_{BM}^{BM}$ reports the percentage gap between the new lower bounds $L_{BM}^{BM}$, described in section 3, and the best solution $H$ provided by our new exact algorithm, i.e. $G_{BM}^{BM} = \frac{H - L_{BM}^{BM}}{H} \times 100$. For F1 and F2, the average percentage gaps only consider the instances where a valid lower bound
Table 3  Comparison among new lower bounds, described in section 3, and lower bounds proposed in the literature.

<table>
<thead>
<tr>
<th>Problem Name</th>
<th>New Lower Bounds</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$G_{diff}$</td>
<td>$G_{h}$</td>
</tr>
<tr>
<td>avgut</td>
<td>2.51</td>
<td>7.99</td>
</tr>
<tr>
<td>gcut</td>
<td>2.98</td>
<td>17.99</td>
</tr>
<tr>
<td>gcut 1-4</td>
<td>3.06</td>
<td>4.58</td>
</tr>
<tr>
<td>gcut 1-12</td>
<td>3.59</td>
<td>7.30</td>
</tr>
<tr>
<td>gcut 1-13</td>
<td>3.74</td>
<td>7.36</td>
</tr>
<tr>
<td>bong</td>
<td>0.00</td>
<td>58.32</td>
</tr>
<tr>
<td>ht</td>
<td>0.00</td>
<td>32.96</td>
</tr>
<tr>
<td>bkw</td>
<td>0.53</td>
<td>34.31</td>
</tr>
</tbody>
</table>

is computed (e.g., for gcut13 and bkw13 F1 is not solved). $L_{BM}$ represents the value of the new overall lower bound after the postprocessing, described in section 3, and $G_{BM}$ is the corresponding percentage gap with respect to $H$. The computing time of $L_{diff}$ and $L_{h}$ are not reported, because they are negligible. While, for the remaining lower bounds the computing time in seconds is reported in column $T$. For $L_{F1}$, column $S$ shows the number of lower bounds solved within the given time limit. While, for $L_{F2}$ column $S$ shows the number of lower bounds computed because all the y-feasible subsets have been generated. Our lower bounds can be compared with the lower bound values $L_{MMV}$ and $L_{BSM}$ reported by Martello et al. (2003) (and also provided by Iori et al. (2003) for set class) and Belov et al. (2006), respectively, and the percentage deviation $G_{APT}$ of the lower bound from the best solution known reported in Alvarez-Valdes et al. (2007). In order to allow a complete comparison, for set gcut we also report the averages computed on instances 1–4 and 1–12.

The results show that our new overall lower bound $L_{BM}$ outperforms the lower bounds proposed in the literature, but often it takes large computing time. However, as the exact algorithm performance demonstrates, in many cases the less expensive lower and upper bounds are sufficient to prove the optimality and only for the hardest instances the computation of $G_{F1}$ and $G_{F2}$ are required and the computing time spent is often repayed.

### 6.3 Upper bounds

Table 4 compares new upper bounds, described in section 4, and upper bounds proposed in the literature.

Columns $G_{x}$ reports the percentage gap between the new upper bounds $H_{x}$, described in section 4, and our best lower bound $\hat{L}$ provided by our exact method, i.e. $G_{x} = \frac{H_{x} - L}{L} \times 100$. The value of the new overall upper bound after the postprocessing, described in section 4, is shown in column $H_{BM}$. The computing time of $H_{NPS}$ and $H_{PBF}$ are in average under the 10 seconds, while the
computing time in seconds of $H_{HF1}$ and $H_{BM}$ are reported in column $T$. For $H_{HF1}$, we also report in column $S$ the number of instances where procedure $HF1$ has found a feasible solution. In order to allow a complete comparison, for sets $gcut$ and $beng$ we also report the averages computed on instances 1–8 and 1–7, respectively.

The new upper bound $H_{BM}$ can be compared with the value of the upper bound $H_{IMM}, H_{LMMM}, H_{BKW}, H_{APT}, H_B$ and $H_{BSM}$ reported by Iori et al. (2003), Burke et al. (2004), Lesh et al. (2005), Alvarez-Valdes et al. (2006), Bortfeldt (2006) and Belov et al. (2006), respectively. The overall upper bound $H_{BM}$ is competitive with respect to $H_{APT}$ and $H_{BSM}$, while it outperforms the other heuristic algorithms proposed in the literature.

### 6.4 Branch and bound algorithm

Table 5 compares the results obtained with the new branch and bound procedure, described in section 5, and the exact methods proposed in the literature by Martello et al. (2003) and Alvarez-Valdes et al. (2007). For each exact algorithm we report the average of the percentage gap $G$ between the best lower bound $L$ and the best feasible solution found $H$, i.e. $G = \frac{H}{H} \times 100$, the maximum percentage gap $G_{\text{max}}$ within the set of test instances, the computing time $T$ in seconds for the test instances solved to optimality, whose number is reported in column $S$. In order to allow a complete comparison, for set $gcut$ we also report the averages computed on instances 1–4.

Martello et al. (2003) have used a Pentium III 800MHz with a time limit of 3600 seconds, while Alvarez-Valdes et al. (2007) have used a Pentium IV 2GHz with a time limit of 1200 seconds. The new exact algorithm, within a time limit of 1200 seconds, solves to optimality more instances than Martello et al. (2003) (8 out of 38) and Alvarez-Valdes et al. (2007) (20 out of 538) and the average and maximum gaps are better. Only for set $\text{Class 10}$ Alvarez-Valdes et al. (2007) obtain a better result. Moreover, the new exact algorithm solves to optimality 10 instances of sets $gcut$ and
Table 5 Comparison among the exact algorithms proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>tG</td>
<td>sG</td>
</tr>
<tr>
<td>ncut</td>
<td>13.31 0.00 0.00</td>
<td>118.35 0.17 2.00</td>
<td>6.97 0.00 0.00 12</td>
</tr>
<tr>
<td>cgcut 1-4</td>
<td>0.00 1.21 2.10</td>
<td>11.48 3.68 5.97 1</td>
<td>0.02 1.87 3.08 1</td>
</tr>
<tr>
<td>gcut 1-13</td>
<td>1.33 0.45 3.46</td>
<td>3 0.00 2.71 6.21 2</td>
<td>0.12 0.14 0.50 2</td>
</tr>
<tr>
<td>beng</td>
<td>138.29 0.00 0.00</td>
<td>336.89 0.65 2.70 6</td>
<td>1.00 0.00 0.00 10</td>
</tr>
<tr>
<td>ht</td>
<td>350.00 0.00 0.00</td>
<td>514.33 0.72 2.31 7</td>
<td>1.35 0.36 3.23 8</td>
</tr>
<tr>
<td>bkw</td>
<td>345.04 0.53 1.23</td>
<td>41.58 0.10 0.47 41</td>
<td></td>
</tr>
<tr>
<td>Class 1</td>
<td>102.00 0.00 0.00</td>
<td>2.66 0.29 2.36 43</td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>32.10 0.47 2.01</td>
<td>64.62 0.60 2.62</td>
<td></td>
</tr>
<tr>
<td>Class 4</td>
<td>608.88 1.86 2.65</td>
<td>8.00 2.19 3.01 1</td>
<td></td>
</tr>
<tr>
<td>Class 5</td>
<td>26.73 0.34 1.82</td>
<td>40.80 0.35 1.89 31</td>
<td></td>
</tr>
<tr>
<td>Class 6</td>
<td>0.76 0.00 0.00</td>
<td>1.32 0.00 0.00 50</td>
<td></td>
</tr>
<tr>
<td>Class 8</td>
<td>37.67 2.95 3.94</td>
<td>10.55 3.12 4.25 1</td>
<td></td>
</tr>
<tr>
<td>Class 9</td>
<td>0.06 0.00 0.00</td>
<td>0.17 0.00 0.00 50</td>
<td></td>
</tr>
</tbody>
</table>

bkw not included in the computational results of Martello et al. (2003) and Alvarez-Valdes et al. (2007). These instances have a large number of items and/or a large strip width. The computing time required by the new exact algorithm is sometime larger than the computing time reported by Alvarez-Valdes et al. (2007), but often we are repaid by solving to optimality more instances. Only for sets ncut, beng, Class 4 and Class 8 we do not take advantages. On the other hand, for many sets we solve the same number or more instances in less computing time.

7. Conclusions

In this paper we have proposed a new exact algorithm for the 2SP that makes use of three reduction procedures, four lower bounds and three upper bounds. The new exact algorithm is a branch and bound that follows the classical depth-search strategy using the proposed reduction procedures, lower and upper bounds and a new rule to generate the tree nodes.

The main contributions of this paper, in our view, include: a new reduction procedure, described in section 2.2.1; three completely new lower bounds, described in sections 3.3, 3.4.1 and 3.4.2; a new postprocessing to improve the lower bound, described in section 3.5; three new heuristic algorithms, described in section 4; a branch and bound with a new rule to generate nodes and a simple dynamic strategy to select the node to expand, described in section 5.

In our computational experiments we have considered a large number of test instance sets, that also include instances not considered in the previous papers, in particular for the exact algorithms proposed in the literature so far. The new exact algorithm solves to optimality several instances not solved so far and the average and maximum gaps are better than the ones reported in the literature (only for set Class 10 Alvarez-Valdes et al. (2007) obtain a better result).

References


Online Supplement to “An Exact Algorithm for the Two-Dimensional Strip Packing Problem”

Marco Antonio Boschetti
Department of Mathematics, University of Bologna, Via Sacchi 3, 47023 Cesena, Italy, marco.boschetti@unibo.it,

Lorenza Montaletti
Department of Mathematics, University of Bologna, Via Sacchi 3, 47023 Cesena, Italy, montaletti.l@libero.it,

The paper “An Exact Algorithm for the Two-Dimensional Strip Packing Problem” gives a detailed description of the algorithm, but only summarize the computational results. This online supplement gives a complete and detailed description of the computational results to better understand the performance of the new lower and upper bounds and of the new exact algorithm.

Subject classifications: Production/scheduling; cutting stock/trim; programming; integer; algorithms: branch and bound.
Area of review: Optimization
History: Sent November 2007

1. Detailed computational results

The algorithms presented in this paper have been coded in ANSI C using Microsoft Visual Studio 6 and run on a Laptop equipped with an Intel Pentium M 725 1.60 GHz. Ilog CPLEX 10.1 was used as mixed integer linear programming solver for computing lower bound \( L_{BMF1} \), described in section 3.4.1, and as linear programming solver for computing lower bound \( L_{BMF2} \), described in section 3.4.2.

We have considered the following set of test problems:

- \( ngcut \) contains 12 instances from Beasley (1985b);
- \( cgcut \) contains 3 instances from Christofides and Whitlock (1977);
- \( gcut \) contains 13 instances from Beasley (1985a);
- \( beng \) contains 10 instances from Bengtsson (1982);
- \( ht \) contains 9 instances from Hopper and Turton (2001);
- \( bkw \) contains 13 instances from Burke et al. (2004);
- \( class \) contains 10 classes. Each class contains 5 groups of 10 instances, that differ for the different number of items. The first 6 classes were proposed by Berkey and Wang (1987), the other 4 classes were proposed by Martello and Vigo (1998).

The two sets \( ht \) and \( bkw \) are obtained by cutting a starting master \((W,H)\) into smaller rectangles, so the optimal solution value is always equal to continuous bound, i.e. equal to \( H \). While, the
remaining sets are well known test instances for other two-dimensional packing problems that have been transformed into strip-packing instances by setting the strip width \( W \) to the width of the master surface or of the bin.

In Tables 1–15 we report for all instances the Name, the number of items \( n \) and the width of the strip \( W \). Tables 1, 2, 4, 5, 8, 9, 12 and 13 report in each row the result of a single instance. While, Tables 3, 6, 7, 10, 11, 14 and 15 report in each row the aggregate (i.e., average, sum, etc.) results of ten instances. When in our tables a cell is empty it means that the value is not available. While, the character “-” is used when the corresponding value cannot be computed (e.g., if no instances have been solved the average computing time for the solved instances is not available).

1.1 Reductions

Tables 1, 2 and 3 reports the details of the test instances and of the reduction test procedures described in section 2.2. In particular, they show for each row the following data:

- \( \bar{H} \) : best height achieved by the new exact methods;
- \( \Delta n\% \) : percentage reduction of the number of items after reductions described in section 2.2.1, i.e.,
  \[ \Delta n\% = \frac{n - n'}{n} \times 100 \]
  where \( n' \) is the number of items to allocate after reductions;
- \( \Delta W\% \) : percentage reduction of the strip width after reductions described in section 2.2.2, i.e.,
  \[ \Delta W\% = \frac{W - W'}{W} \times 100 \]
  where \( W' \) is the strip width after reductions;
- \( \Delta A\% \) : percentage increasing of the item areas after reductions described in section 2.2.3, i.e.,
  \[ \Delta A\% = \frac{A_f - A'_f}{A_f} \times 100 \]
  where \( A_f \) and \( A'_f \) are the total area of items before and after reductions, respectively;
- \( \Delta H\% \) : percentage of height of the strip containing items fixed in the solution after reduction described in section 2.2.1, i.e.,
  \[ \Delta H\% = \frac{H' - H}{H} \times 100 \]

1.2 Lower bounds

Tables 4, 5, 6 and 7 compare new lower bounds, described in section 3, and lower bounds proposed in the literature. For each row they report:

- \( G_{BM}^{BM}, G_h^{BM}, G_{F1}^{BM}, G_{F2}^{BM}, G_{BM} \) : percentage gap between the new lower bounds \( L_{BM}^{BM} \), described in section 3, and our best solution \( \bar{H} \) provided by our exact method, i.e.
  \[ G_{BM}^{BM} = \frac{\bar{H} - L_{BM}^{BM}}{\bar{H}} \times 100 \]
- \( L_{BM} \) : value of the new overall lower bound after the postprocessing, described in section 3;
- \( T \) : computing time in seconds. The computing time of \( L_{FM}^{BM} \) and \( L_{BM}^{BM} \) are not reported, because they are negligible;
- \( S \) : for \( L_{FM}^{BM} \) is the number of lower bounds solved within the given time limit and for \( L_{BM}^{BM} \) is the number of lower bounds computed because all the \( y \)-feasible subsets have been generated;
The results show that our new overall lower bound $L_{BM}$ outperforms the lower bounds proposed in the literature, even if for some instances $L_{F1}^{BM}$ is time consuming and/or $L_{F2}^{BM}$ cannot be computed.

### 1.3 Upper bounds

Tables 8, 9, 10 and 11 compare new upper bounds, described in section 4, and upper bounds proposed in the literature. For each row they report:

- $G_{NPS}, G_{PBF}, G_{HF1}, G_{BM}$: percentage gap between the new upper bounds $H_x$, described in section 4, and our best lower bound $L$ provided by our exact method, i.e. $G_x = \frac{H_x-L}{L} \times 100$;
- $H_{BM}$: value of our overall upper bound after the postprocessing, described in section 4;
- $H_{IMM}, H_{LMMM}, H_{BKW}, H_{APT}, H_{B}, H_{BSM}$: values of the upper bounds proposed by Iori et al. (2003), Burke et al. (2004), Lesh et al. (2005), Alvarez-Valdes et al. (2006), Bortfeldt (2006) and Belov et al. (2006), respectively;
- $T$: computing time in seconds.

### 1.4 Branch and bound algorithm

Tables 12, 13, 14 and 15 compare the results obtained with the new branch and bound procedure, described in section 5, and the exact methods proposed in the literature by Martello et al. (2003) and Alvarez-Valdes et al. (2007). For each row they report:

- $L, H$: lower and upper bound at the end of the branch and bound. If the optimum solution is reached within the given time limit $L = H$;
- $G$: percentage gap between the lower bound $L$ and the best solution found $H$, i.e. $\text{Gap} = \frac{H-L}{H} \times 100$;
- $G_{\text{max}}$: maximum percentage gap between the lower bound $L$ and the best solution found $H$ within the test instances of the corresponding group;
- $T$: computing time in seconds for the test instances solved to optimality;
- $S$: number of test instances solved to optimality within the given time limit.
### Table 1  Test instances *ngcut*, *cgcut* and *gcut*: reduction test procedures.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Reductions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>n</strong></td>
</tr>
<tr>
<td>ngcut1</td>
<td>10</td>
</tr>
<tr>
<td>ngcut2</td>
<td>17</td>
</tr>
<tr>
<td>ngcut3</td>
<td>21</td>
</tr>
<tr>
<td>ngcut4</td>
<td>7</td>
</tr>
<tr>
<td>ngcut5</td>
<td>14</td>
</tr>
<tr>
<td>ngcut6</td>
<td>15</td>
</tr>
<tr>
<td>ngcut7</td>
<td>8</td>
</tr>
<tr>
<td>ngcut8</td>
<td>13</td>
</tr>
<tr>
<td>ngcut9</td>
<td>18</td>
</tr>
<tr>
<td>ngcut10</td>
<td>13</td>
</tr>
<tr>
<td>ngcut11</td>
<td>15</td>
</tr>
<tr>
<td>ngcut12</td>
<td>22</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
</tr>
<tr>
<td>cgcut1</td>
<td>16</td>
</tr>
<tr>
<td>cgcut2</td>
<td>23</td>
</tr>
<tr>
<td>cgcut3</td>
<td>62</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
</tr>
<tr>
<td>gcut1</td>
<td>10</td>
</tr>
<tr>
<td>gcut2</td>
<td>20</td>
</tr>
<tr>
<td>gcut3</td>
<td>50</td>
</tr>
<tr>
<td>gcut4</td>
<td>50</td>
</tr>
<tr>
<td>gcut5</td>
<td>10</td>
</tr>
<tr>
<td>gcut6</td>
<td>20</td>
</tr>
<tr>
<td>gcut7</td>
<td>30</td>
</tr>
<tr>
<td>gcut8</td>
<td>50</td>
</tr>
<tr>
<td>gcut9</td>
<td>10</td>
</tr>
<tr>
<td>gcut10</td>
<td>20</td>
</tr>
<tr>
<td>gcut11</td>
<td>30</td>
</tr>
<tr>
<td>gcut12</td>
<td>50</td>
</tr>
<tr>
<td>gcut13</td>
<td>32</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td></td>
</tr>
</tbody>
</table>
Table 2  Test instances beng, ht and bkw: reduction test procedures.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Reductions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>n</td>
</tr>
<tr>
<td>beng1</td>
<td>20</td>
</tr>
<tr>
<td>beng2</td>
<td>40</td>
</tr>
<tr>
<td>beng3</td>
<td>60</td>
</tr>
<tr>
<td>beng4</td>
<td>80</td>
</tr>
<tr>
<td>beng5</td>
<td>100</td>
</tr>
<tr>
<td>beng6</td>
<td>40</td>
</tr>
<tr>
<td>beng7</td>
<td>80</td>
</tr>
<tr>
<td>beng8</td>
<td>120</td>
</tr>
<tr>
<td>beng9</td>
<td>160</td>
</tr>
<tr>
<td>beng10</td>
<td>200</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>0.00</td>
</tr>
<tr>
<td>ht1</td>
<td>16</td>
</tr>
<tr>
<td>ht2</td>
<td>17</td>
</tr>
<tr>
<td>ht3</td>
<td>16</td>
</tr>
<tr>
<td>ht4</td>
<td>25</td>
</tr>
<tr>
<td>ht5</td>
<td>25</td>
</tr>
<tr>
<td>ht6</td>
<td>25</td>
</tr>
<tr>
<td>ht7</td>
<td>28</td>
</tr>
<tr>
<td>ht8</td>
<td>29</td>
</tr>
<tr>
<td>ht9</td>
<td>28</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>0.00</td>
</tr>
<tr>
<td>bkw1</td>
<td>10</td>
</tr>
<tr>
<td>bkw2</td>
<td>20</td>
</tr>
<tr>
<td>bkw3</td>
<td>30</td>
</tr>
<tr>
<td>bkw4</td>
<td>40</td>
</tr>
<tr>
<td>bkw5</td>
<td>50</td>
</tr>
<tr>
<td>bkw6</td>
<td>60</td>
</tr>
<tr>
<td>bkw7</td>
<td>70</td>
</tr>
<tr>
<td>bkw8</td>
<td>80</td>
</tr>
<tr>
<td>bkw9</td>
<td>100</td>
</tr>
<tr>
<td>bkw10</td>
<td>200</td>
</tr>
<tr>
<td>bkw11</td>
<td>200</td>
</tr>
<tr>
<td>bkw12</td>
<td>500</td>
</tr>
<tr>
<td>bkw13</td>
<td>3152</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>8.19</td>
</tr>
</tbody>
</table>
Table 3  Test instances class: reduction test procedures.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Name</th>
<th>n W H</th>
<th>$\Delta n%$</th>
<th>$\Delta W%$</th>
<th>$\Delta A%$</th>
<th>$\Delta H%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>20 10 61.2</td>
<td>42.00 0.00 0.17 48.48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 10 121.8</td>
<td>61.25 0.00 0.00 64.62</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 10 188.5</td>
<td>50.67 0.00 0.00 58.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 10 262.6</td>
<td>71.63 0.00 0.00 76.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 10 304.8</td>
<td>41.60 0.00 0.00 48.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>53.43 0.00 0.03 59.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td>20 30 19.7</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 30 39.1</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 30 60.1</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 30 83.2</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 30 100.5</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>20 40 162.6</td>
<td>26.50 0.00 1.34 27.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 40 333.8</td>
<td>53.50 0.00 0.15 54.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 40 506.3</td>
<td>42.17 0.00 0.05 41.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 40 679.8</td>
<td>71.63 0.00 0.00 76.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 40 838.7</td>
<td>50.20 0.00 0.06 50.37</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>44.87 0.00 0.33 45.37</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 4</td>
<td>20 100 63.8</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100 126.3</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100 196.4</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100 271.3</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100 325.7</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 5</td>
<td>20 100 534.1</td>
<td>54.50 0.00 1.73 58.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100 1073.8</td>
<td>70.25 0.00 0.00 72.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100 1644.0</td>
<td>65.17 0.00 0.00 68.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100 2288.5</td>
<td>72.25 0.00 0.00 75.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100 2646.3</td>
<td>57.30 0.00 0.00 59.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>63.89 0.00 0.59 66.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 6</td>
<td>20 300 168.1</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 300 334.3</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 300 521.1</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 300 719.4</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 300 864.0</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>0.00 0.00 0.00 0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 7</td>
<td>20 100 501.9</td>
<td>85.50 0.30 7.00 87.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100 1059.0</td>
<td>92.00 0.00 4.58 90.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100 1529.6</td>
<td>94.00 0.00 1.41 93.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100 2198.6</td>
<td>99.63 0.00 1.67 99.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>91.80 0.06 3.10 90.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 8</td>
<td>20 100 457.6</td>
<td>1.50 0.00 0.08 2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100 952.6</td>
<td>3.50 0.00 0.05 3.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100 1401.6</td>
<td>0.83 0.00 0.00 0.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100 1966.8</td>
<td>0.37 0.00 0.00 0.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100 2429.5</td>
<td>1.00 0.00 0.01 1.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>1.44 0.00 0.03 1.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 9</td>
<td>20 100 1106.8</td>
<td>100.00 0.20 8.06 100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100 2190.6</td>
<td>96.50 0.10 5.04 97.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100 3410.4</td>
<td>100.00 0.00 2.97 100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100 4588.1</td>
<td>99.63 0.00 1.37 99.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100 5431.9</td>
<td>99.70 0.00 0.66 99.71</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>99.17 0.06 3.62 99.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 10</td>
<td>20 100 351.1</td>
<td>20.00 0.00 0.68 28.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100 665.0</td>
<td>23.00 0.00 0.33 28.27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100 937.0</td>
<td>19.83 0.00 0.22 23.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100 1209.3</td>
<td>11.38 0.00 0.01 17.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100 1511.9</td>
<td>8.30 0.00 0.01 13.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Avg</td>
<td>16.50 0.00 0.25 22.26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 4

Test instances \(ngcut\), \(cgcut\) and \(gcut\): comparison among lower bounds proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Lower Bounds</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(G_{BM}^{\text{MMV}})</td>
<td>(T)</td>
</tr>
<tr>
<td>(ngcut1)</td>
<td>10</td>
<td>13.04</td>
</tr>
<tr>
<td>(ngcut2)</td>
<td>10</td>
<td>3.33</td>
</tr>
<tr>
<td>(ngcut3)</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>(ngcut4)</td>
<td>10</td>
<td>10.00</td>
</tr>
<tr>
<td>(ngcut5)</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>(ngcut6)</td>
<td>10</td>
<td>6.45</td>
</tr>
<tr>
<td>(ngcut7)</td>
<td>20</td>
<td>25.00</td>
</tr>
<tr>
<td>(ngcut8)</td>
<td>20</td>
<td>3.03</td>
</tr>
<tr>
<td>(ngcut9)</td>
<td>20</td>
<td>2.00</td>
</tr>
<tr>
<td>(ngcut10)</td>
<td>30</td>
<td>27.50</td>
</tr>
<tr>
<td>(ngcut11)</td>
<td>30</td>
<td>3.85</td>
</tr>
<tr>
<td>(ngcut12)</td>
<td>30</td>
<td>3.45</td>
</tr>
</tbody>
</table>

| Avg | 8.14 | 12.05 | 0.00 | 34.63 | 12 | 5.68 | 0.02 | 12 | 0.00 | 40.83 | 34.64 | 40.75 | 0.17 |

| \(cgcut1\) | 10 | 0.00 | 21.74 | 0.00 | 0.09 | 1 | 0.00 | 0.13 | 1 | 0.00 | 23 | 0.22 |
| \(cgcut2\) | 70 | 3.08 | 29.23 | 3.08 | 664.34 | 0 | 3.08 | 1.63 | 1 | 1.54 | 64 | 673.78 |
| \(cgcut3\) | 70 | 3.75 | 3.00 | 2.10 | 660.24 | 0 | 2.25 | 1.30 | 1 | 2.10 | 653 | 662.08 |

| Avg | 2.28 | 17.99 | 1.73 | 441.56 | 1 | 1.78 | 1.02 | 3 | 1.21 | 246.67 | 445.36 | 240.67 | 1.38 |

| \(gcut\) | 10 | 0.00 | 0.00 | 0.00 | 0.00 | 1 | 0.00 | 0.00 | 1 | 0.00 | 1016 | 0.00 |
| \(gcut2\) | 20 | 2.36 | 8.09 | 0.00 | 6.63 | 1 | 0.25 | 0.02 | 1 | 0.00 | 1187 | 6.64 |
| \(gcut3\) | 30 | 250 | 0.00 | 0.00 | 0.00 | 1 | 0.00 | 0.00 | 1 | 0.00 | 1803 | 0.00 |
| \(gcut4\) | 50 | 11.0 | 10.22 | 0.33 | 660.20 | 0 | 0.40 | 0.02 | 1 | 0.00 | 2993 | 661.05 |
| \(gcut5\) | 50 | 4.95 | 13.98 | 0.00 | 0.58 | 1 | 4.01 | 0.00 | 1 | 0.00 | 1273 | 0.58 |
| \(gcut6\) | 50 | 1.45 | 8.54 | 0.00 | 0.55 | 1 | 1.03 | 0.00 | 1 | 0.00 | 2622 | 0.55 |
| \(gcut7\) | 50 | 500 | 0.00 | 2.34 | 0.00 | 0.30 | 1 | 0.00 | 0.02 | 1 | 0.00 | 4093 | 0.31 |
| \(gcut8\) | 50 | 2.10 | 17.01 | 1.46 | 661.52 | 0 | 1.46 | 0.05 | 1 | 1.46 | 5809 | 662.64 |
| \(gcut9\) | 1000 | 3.50 | 9.06 | 0.00 | 0.02 | 1 | 3.50 | 0.02 | 1 | 0.00 | 2317 | 0.03 |
| \(gcut10\) | 1000 | 1.37 | 5.11 | 0.00 | 1.53 | 1 | 0.91 | 0.00 | 1 | 0.00 | 5964 | 1.53 |
| \(gcut11\) | 1000 | 2.24 | 13.22 | 0.76 | 660.63 | 0 | 0.84 | 0.02 | 1 | 0.76 | 6826 | 661.24 |
| \(gcut12\) | 1000 | 0.00 | 0.00 | 0.00 | 0.00 | 1 | 0.00 | 0.00 | 1 | 0.00 | 14690 | 0.00 |
| \(gcut13\) | 3000 | 3.54 | 8.13 | - | 486.44 | 0 | 3.46 | 22.28 | 1 | 3.46 | 4776 | 509.19 |

| Avg | 1.74 | 7.30 | 0.21 | 165.99 | 9 | 1.22 | 1.72 | 3 | 1.37 | 4305.31 | 192.60 |
| \(Avg_{12}\) | 1.59 | 7.30 | 0.21 | 165.99 | 9 | 1.03 | 0.01 | 12 | 0.21 | 4266.08 | 166.21 | 0.31 |
| \(Avg_{14}\) | 0.86 | 4.58 | 0.08 | 166.71 | 3 | 0.16 | 0.01 | 4 | 0.08 | 1749.75 | 166.92 | 1721.50 |
Table 5  Test instances *beng, ht* and *bkw* comparison among lower bounds proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Lower Bounds</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Name</em></td>
<td><em>n</em></td>
</tr>
<tr>
<td></td>
<td>beng1</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>beng2</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>beng3</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>beng4</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>beng5</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>beng6</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>beng7</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>beng8</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>beng9</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>beng10</td>
<td>200</td>
</tr>
<tr>
<td>Avg</td>
<td>0.00</td>
<td>58.32</td>
</tr>
<tr>
<td></td>
<td>ht1</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>ht2</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>ht3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>ht4</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>ht5</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>ht6</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>ht7</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>ht8</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>ht9</td>
<td>28</td>
</tr>
<tr>
<td>Avg</td>
<td>0.00</td>
<td>32.90</td>
</tr>
<tr>
<td></td>
<td>bkw1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>bkw2</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>bkw3</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>bkw4</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>bkw5</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>bkw6</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>bkw7</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>bkw8</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>bkw9</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>bkw10</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>bkw11</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>bkw12</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>bkw13</td>
<td>3152</td>
</tr>
<tr>
<td>Avg</td>
<td>0.53</td>
<td>34.31</td>
</tr>
<tr>
<td>Problem</td>
<td>Name</td>
<td>W</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>---</td>
</tr>
<tr>
<td>Class 1</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class 2</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class 3</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class 4</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class 5</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6** Test instances class: comparison among lower bounds proposed in this paper and in the literature.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Test instances class</th>
<th>Description</th>
<th>New Lower Bounds</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 6</td>
<td>20 300</td>
<td></td>
<td>(L_{BM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 300</td>
<td></td>
<td>(L_{BSM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 300</td>
<td></td>
<td>(G_{L,B}^M )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 300</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 300</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>(L_{MMY} )</td>
<td></td>
</tr>
<tr>
<td>Class 7</td>
<td>20 100</td>
<td></td>
<td>(L_{BM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100</td>
<td></td>
<td>(L_{BSM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100</td>
<td></td>
<td>(G_{L,B}^M )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>(L_{MMY} )</td>
<td></td>
</tr>
<tr>
<td>Class 8</td>
<td>20 100</td>
<td></td>
<td>(L_{BM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100</td>
<td></td>
<td>(L_{BSM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100</td>
<td></td>
<td>(G_{L,B}^M )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>(L_{MMY} )</td>
<td></td>
</tr>
<tr>
<td>Class 9</td>
<td>20 100</td>
<td></td>
<td>(L_{BM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100</td>
<td></td>
<td>(L_{BSM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100</td>
<td></td>
<td>(G_{L,B}^M )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>(L_{MMY} )</td>
<td></td>
</tr>
<tr>
<td>Class 10</td>
<td>20 100</td>
<td></td>
<td>(L_{BM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 100</td>
<td></td>
<td>(L_{BSM} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60 100</td>
<td></td>
<td>(G_{L,B}^M )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100 100</td>
<td></td>
<td>(G_{L,B}^{10} )</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>(L_{MMY} )</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Test instances class: comparison among lower bounds proposed in this paper and in the literature.
Table 8  Test instances *ngcut*, *cgcut* and *gcut*: comparison among upper bounds proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Heuristics</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>W</td>
<td>(G_{NPS})</td>
</tr>
<tr>
<td>ngcut1</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut2</td>
<td>17</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut3</td>
<td>21</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut4</td>
<td>7</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut5</td>
<td>14</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut6</td>
<td>15</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut7</td>
<td>8</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut8</td>
<td>13</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut9</td>
<td>18</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut10</td>
<td>13</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut11</td>
<td>15</td>
<td>0.00</td>
</tr>
<tr>
<td>ngcut12</td>
<td>22</td>
<td>0.00</td>
</tr>
<tr>
<td>Avg</td>
<td>0.33</td>
<td>0.70</td>
</tr>
<tr>
<td>cgcut1</td>
<td>16</td>
<td>0.00</td>
</tr>
<tr>
<td>cgcut2</td>
<td>23</td>
<td>12.50</td>
</tr>
<tr>
<td>cgcut3</td>
<td>62</td>
<td>3.52</td>
</tr>
<tr>
<td>Avg</td>
<td>5.34</td>
<td>1.81</td>
</tr>
<tr>
<td>gcut1</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>gcut2</td>
<td>20</td>
<td>0.02</td>
</tr>
<tr>
<td>gcut3</td>
<td>30</td>
<td>0.00</td>
</tr>
<tr>
<td>gcut4</td>
<td>50</td>
<td>6.01</td>
</tr>
<tr>
<td>gcut5</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>gcut6</td>
<td>20</td>
<td>2.36</td>
</tr>
<tr>
<td>gcut7</td>
<td>30</td>
<td>5.84</td>
</tr>
<tr>
<td>gcut8</td>
<td>50</td>
<td>6.67</td>
</tr>
<tr>
<td>gcut9</td>
<td>10</td>
<td>0.00</td>
</tr>
<tr>
<td>gcut10</td>
<td>20</td>
<td>0.55</td>
</tr>
<tr>
<td>gcut11</td>
<td>30</td>
<td>8.60</td>
</tr>
<tr>
<td>gcut12</td>
<td>50</td>
<td>6.67</td>
</tr>
<tr>
<td>gcut13</td>
<td>32</td>
<td>10.62</td>
</tr>
<tr>
<td>Avg</td>
<td>3.28</td>
<td>0.81</td>
</tr>
<tr>
<td>Avg_{1-8}</td>
<td>2.86</td>
<td>0.37</td>
</tr>
</tbody>
</table>
### Table 9

Test instances *beng, ht* and *bkw*: comparison among upper bounds proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Heuristics</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name n W</td>
<td>G\text{NPS}</td>
</tr>
<tr>
<td>beng1</td>
<td>20 25</td>
<td>3.33 0.00</td>
</tr>
<tr>
<td>beng2</td>
<td>40 25</td>
<td>3.51 0.00</td>
</tr>
<tr>
<td>beng3</td>
<td>60 25</td>
<td>2.38 0.00</td>
</tr>
<tr>
<td>beng4</td>
<td>80 25</td>
<td>1.87 0.00</td>
</tr>
<tr>
<td>beng5</td>
<td>100 25</td>
<td>0.75 0.00</td>
</tr>
<tr>
<td>beng6</td>
<td>40 40</td>
<td>2.78 0.00</td>
</tr>
<tr>
<td>beng7</td>
<td>80 40</td>
<td>1.49 0.00</td>
</tr>
<tr>
<td>beng8</td>
<td>120 40</td>
<td>0.99 0.00</td>
</tr>
<tr>
<td>beng9</td>
<td>160 40</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>beng10</td>
<td>200 40</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>Avg</td>
<td>1.71 0.00</td>
<td>7.78 53.37</td>
</tr>
<tr>
<td>Avg</td>
<td>2.30 0.00</td>
<td>7.78 50.05</td>
</tr>
<tr>
<td>ht1</td>
<td>16 20</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>ht2</td>
<td>17 20</td>
<td>0.00 5.00</td>
</tr>
<tr>
<td>ht3</td>
<td>16 20</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>ht4</td>
<td>25 40</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>ht5</td>
<td>25 40</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>ht6</td>
<td>25 40</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>ht7</td>
<td>28 60</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>ht8</td>
<td>29 60</td>
<td>0.00 3.33</td>
</tr>
<tr>
<td>ht9</td>
<td>28 60</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>Avg</td>
<td>0.00 0.93</td>
<td>0.00 1.88</td>
</tr>
<tr>
<td>bkw1</td>
<td>10 40</td>
<td>0.00 0.00</td>
</tr>
<tr>
<td>bkw2</td>
<td>20 30</td>
<td>0.00 4.00</td>
</tr>
<tr>
<td>bkw3</td>
<td>30 30</td>
<td>0.00 6.00</td>
</tr>
<tr>
<td>bkw4</td>
<td>40 80</td>
<td>2.50 1.25</td>
</tr>
<tr>
<td>bkw5</td>
<td>50 100</td>
<td>7.00 1.00</td>
</tr>
<tr>
<td>bkw6</td>
<td>60 50</td>
<td>3.00 1.00</td>
</tr>
<tr>
<td>bkw7</td>
<td>70 80</td>
<td>1.00 0.00</td>
</tr>
<tr>
<td>bkw8</td>
<td>80 100</td>
<td>5.00 1.25</td>
</tr>
<tr>
<td>bkw9</td>
<td>100 50</td>
<td>4.67 0.67</td>
</tr>
<tr>
<td>bkw10</td>
<td>200 70</td>
<td>2.00 0.67</td>
</tr>
<tr>
<td>bkw11</td>
<td>200 70</td>
<td>2.00 0.67</td>
</tr>
<tr>
<td>bkw12</td>
<td>500 100</td>
<td>2.67 0.33</td>
</tr>
<tr>
<td>bkw13</td>
<td>3152 640</td>
<td>0.10 0.10</td>
</tr>
<tr>
<td>Avg</td>
<td>3.07 0.69</td>
<td>1.67 46.68</td>
</tr>
</tbody>
</table>
### Table 10: Test instances class comparison among upper bounds proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Heuristics</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$G_{HPS}$</td>
<td>$G_{HPS}$</td>
</tr>
<tr>
<td>Class 1</td>
<td>0.62</td>
<td>0.63</td>
</tr>
<tr>
<td>40 10</td>
<td>1.15</td>
<td>0.00</td>
</tr>
<tr>
<td>60 10</td>
<td>0.94</td>
<td>0.05</td>
</tr>
<tr>
<td>80 10</td>
<td>0.27</td>
<td>0.00</td>
</tr>
<tr>
<td>100 10</td>
<td>0.61</td>
<td>0.03</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>0.72</td>
<td>0.14</td>
</tr>
<tr>
<td>Class 2</td>
<td>2.93</td>
<td>0.59</td>
</tr>
<tr>
<td>40 30</td>
<td>4.15</td>
<td>0.00</td>
</tr>
<tr>
<td>60 30</td>
<td>1.97</td>
<td>0.00</td>
</tr>
<tr>
<td>80 30</td>
<td>1.80</td>
<td>0.00</td>
</tr>
<tr>
<td>100 30</td>
<td>0.95</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>2.36</td>
<td>0.12</td>
</tr>
<tr>
<td>Class 3</td>
<td>1.63</td>
<td>2.46</td>
</tr>
<tr>
<td>40 40</td>
<td>3.75</td>
<td>1.04</td>
</tr>
<tr>
<td>60 40</td>
<td>4.21</td>
<td>0.66</td>
</tr>
<tr>
<td>80 40</td>
<td>2.70</td>
<td>0.35</td>
</tr>
<tr>
<td>100 40</td>
<td>2.04</td>
<td>0.50</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>2.86</td>
<td>1.00</td>
</tr>
<tr>
<td>Class 4</td>
<td>5.18</td>
<td>3.31</td>
</tr>
<tr>
<td>40 100</td>
<td>9.62</td>
<td>2.08</td>
</tr>
<tr>
<td>60 100</td>
<td>8.69</td>
<td>1.74</td>
</tr>
<tr>
<td>80 100</td>
<td>7.07</td>
<td>1.52</td>
</tr>
<tr>
<td>100 100</td>
<td>2.47</td>
<td>0.50</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>7.13</td>
<td>1.96</td>
</tr>
<tr>
<td>Class 5</td>
<td>0.92</td>
<td>1.64</td>
</tr>
<tr>
<td>40 100</td>
<td>2.50</td>
<td>0.57</td>
</tr>
<tr>
<td>60 100</td>
<td>2.07</td>
<td>0.74</td>
</tr>
<tr>
<td>80 100</td>
<td>1.42</td>
<td>0.20</td>
</tr>
<tr>
<td>100 100</td>
<td>2.47</td>
<td>0.50</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>1.88</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Table 11: Test instances class comparison among upper bounds proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Heuristics</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>n</td>
</tr>
<tr>
<td>Class 6</td>
<td>20</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>Avg</td>
<td>8.64</td>
<td>3.20</td>
</tr>
<tr>
<td>Class 7</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Avg</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>Class 8</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Avg</td>
<td>7.53</td>
<td>3.92</td>
</tr>
<tr>
<td>Class 9</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Avg</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Class 10</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Avg</td>
<td>4.37</td>
<td>2.03</td>
</tr>
</tbody>
</table>
Table 12  Test instances *ngcut*, *cgcut* and *gcut*: comparison among the branch and bound procedures proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ngcut1</td>
<td>10</td>
<td>10</td>
<td>23</td>
<td>23</td>
<td>0.08</td>
<td>0.00</td>
<td>23 23 0.05 0.00</td>
<td>23 23 2.11 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut2</td>
<td>17</td>
<td>10</td>
<td>30</td>
<td>30</td>
<td>0.47</td>
<td>0.00</td>
<td>30 30 11.31 0.00</td>
<td>30 30 3.11 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut3</td>
<td>21</td>
<td>10</td>
<td>28</td>
<td>28</td>
<td>1.62</td>
<td>0.00</td>
<td>28 28 27.01 0.00</td>
<td>28 28 0.00 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut4</td>
<td>7</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>0.14</td>
<td>0.00</td>
<td>20 20 0.00 0.00</td>
<td>20 20 0.03 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut5</td>
<td>14</td>
<td>10</td>
<td>36</td>
<td>36</td>
<td>0.34</td>
<td>0.00</td>
<td>36 36 0.00 0.00</td>
<td>36 36 0.00 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut6</td>
<td>15</td>
<td>10</td>
<td>31</td>
<td>31</td>
<td>0.84</td>
<td>0.00</td>
<td>31 31 727.20 0.00</td>
<td>31 31 4.39 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut7</td>
<td>8</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>0.38</td>
<td>0.00</td>
<td>20 20 0.00 0.00</td>
<td>20 20 0.00 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut8</td>
<td>13</td>
<td>20</td>
<td>28</td>
<td>28</td>
<td>6.58</td>
<td>0.00</td>
<td>28 28 27.01 0.00</td>
<td>28 28 0.00 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut9</td>
<td>18</td>
<td>20</td>
<td>50</td>
<td>50</td>
<td>0.14</td>
<td>0.00</td>
<td>50 50 T.L. 2.00</td>
<td>50 50 56.13 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut10</td>
<td>13</td>
<td>30</td>
<td>80</td>
<td>80</td>
<td>6.58</td>
<td>0.00</td>
<td>80 80 0.00 0.00</td>
<td>80 80 2.61 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut11</td>
<td>15</td>
<td>30</td>
<td>52</td>
<td>52</td>
<td>107.47</td>
<td></td>
<td>52 52 483.01 0.00</td>
<td>52 52 11.88 0.00</td>
<td></td>
</tr>
<tr>
<td>ngcut12</td>
<td>22</td>
<td>30</td>
<td>87</td>
<td>87</td>
<td>2.11</td>
<td>0.00</td>
<td>87 87 0.00 0.00</td>
<td>87 87 0.00 0.00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avg</th>
<th>40.83</th>
<th>40.83</th>
<th>35.11</th>
<th>0.00</th>
<th>40.75</th>
<th>40.83</th>
<th>118.35</th>
<th>0.17</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgcut1</td>
<td>16</td>
<td>10</td>
<td>23</td>
<td>23</td>
<td>0.69</td>
<td>0.00</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>cgcut2</td>
<td>23</td>
<td>70</td>
<td>64</td>
<td>64</td>
<td>3528.55</td>
<td>0.54</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>cgcut3</td>
<td>62</td>
<td>70</td>
<td>536</td>
<td>536</td>
<td>1200.00</td>
<td>2.10</td>
<td>536</td>
<td>536</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avg</th>
<th>246.67</th>
<th>251.67</th>
<th>0.00</th>
<th>1.21</th>
<th>246.67</th>
<th>253.33</th>
<th>11.48</th>
<th>3.68</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcut1</td>
<td>10</td>
<td>250</td>
<td>1016</td>
<td>1016</td>
<td>0.00</td>
<td>0.00</td>
<td>1016</td>
<td>1016</td>
</tr>
<tr>
<td>gcut2</td>
<td>20</td>
<td>250</td>
<td>1187</td>
<td>1187</td>
<td>7.41</td>
<td>0.00</td>
<td>1187</td>
<td>1187</td>
</tr>
<tr>
<td>gcut3</td>
<td>30</td>
<td>250</td>
<td>1803</td>
<td>1803</td>
<td>0.00</td>
<td>0.00</td>
<td>1803</td>
<td>1803</td>
</tr>
<tr>
<td>gcut4</td>
<td>50</td>
<td>250</td>
<td>2933</td>
<td>2933</td>
<td>1200.01</td>
<td>0.33</td>
<td>2933</td>
<td>2933</td>
</tr>
<tr>
<td>gcut5</td>
<td>10</td>
<td>500</td>
<td>1273</td>
<td>1273</td>
<td>0.69</td>
<td>0.00</td>
<td>1273</td>
<td>1273</td>
</tr>
<tr>
<td>gcut6</td>
<td>20</td>
<td>500</td>
<td>2622</td>
<td>2622</td>
<td>1.33</td>
<td>0.00</td>
<td>2622</td>
<td>2622</td>
</tr>
<tr>
<td>gcut7</td>
<td>30</td>
<td>500</td>
<td>4693</td>
<td>4693</td>
<td>0.36</td>
<td>0.00</td>
<td>4693</td>
<td>4693</td>
</tr>
<tr>
<td>gcut8</td>
<td>50</td>
<td>500</td>
<td>5814</td>
<td>5895</td>
<td>1200.02</td>
<td>1.37</td>
<td>5814</td>
<td>5895</td>
</tr>
<tr>
<td>gcut9</td>
<td>10</td>
<td>1000</td>
<td>2317</td>
<td>2317</td>
<td>0.09</td>
<td>0.00</td>
<td>2317</td>
<td>2317</td>
</tr>
<tr>
<td>gcut10</td>
<td>20</td>
<td>1000</td>
<td>5964</td>
<td>5964</td>
<td>2.08</td>
<td>0.00</td>
<td>5964</td>
<td>5964</td>
</tr>
<tr>
<td>gcut11</td>
<td>30</td>
<td>1000</td>
<td>6834</td>
<td>6878</td>
<td>1200.08</td>
<td>0.64</td>
<td>6834</td>
<td>6878</td>
</tr>
<tr>
<td>gcut12</td>
<td>50</td>
<td>1000</td>
<td>14690</td>
<td>14690</td>
<td>0.00</td>
<td>0.00</td>
<td>14690</td>
<td>14690</td>
</tr>
<tr>
<td>gcut13</td>
<td>32</td>
<td>3000</td>
<td>4776</td>
<td>4947</td>
<td>1201.26</td>
<td>3.46</td>
<td>4776</td>
<td>4947</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avg</th>
<th>4306.31</th>
<th>4329.85</th>
<th>1.33</th>
<th>0.45</th>
<th>1749.75</th>
<th>1752.25</th>
<th>2.47</th>
<th>0.08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>1749.75</td>
<td>1752.25</td>
<td>2.47</td>
<td>0.08</td>
<td>1721.50</td>
<td>1776.00</td>
<td>0.00</td>
<td>2.71</td>
</tr>
<tr>
<td>Avg</td>
<td>1748.75</td>
<td>1753.00</td>
<td>0.12</td>
<td>0.14</td>
<td>1748.75</td>
<td>1753.00</td>
<td>0.12</td>
<td>0.14</td>
</tr>
</tbody>
</table>
Table 13  Test instances *beng*, *ht* and *bkw*: comparison among the branch and bound procedures proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>n</strong></td>
<td><strong>W</strong></td>
<td><strong>L</strong></td>
</tr>
<tr>
<td>beng1</td>
<td>20</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>beng2</td>
<td>40</td>
<td>25</td>
<td>57</td>
</tr>
<tr>
<td>beng3</td>
<td>60</td>
<td>25</td>
<td>84</td>
</tr>
<tr>
<td>beng4</td>
<td>80</td>
<td>25</td>
<td>107</td>
</tr>
<tr>
<td>beng5</td>
<td>100</td>
<td>25</td>
<td>134</td>
</tr>
<tr>
<td>beng6</td>
<td>40</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>beng7</td>
<td>120</td>
<td>40</td>
<td>101</td>
</tr>
<tr>
<td>beng8</td>
<td>160</td>
<td>40</td>
<td>126</td>
</tr>
<tr>
<td>beng9</td>
<td>160</td>
<td>40</td>
<td>126</td>
</tr>
<tr>
<td>beng10</td>
<td>200</td>
<td>40</td>
<td>156</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>89.80</td>
</tr>
<tr>
<td>ht1</td>
<td>16</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>ht2</td>
<td>17</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>ht3</td>
<td>16</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>ht4</td>
<td>25</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>ht5</td>
<td>25</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>ht6</td>
<td>25</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>ht7</td>
<td>28</td>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>ht8</td>
<td>29</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>ht9</td>
<td>28</td>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>21.67</td>
</tr>
<tr>
<td>bkw1</td>
<td>10</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>bkw2</td>
<td>20</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>bkw3</td>
<td>30</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>bkw4</td>
<td>40</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>bkw5</td>
<td>50</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>bkw6</td>
<td>60</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>bkw7</td>
<td>70</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>bkw8</td>
<td>80</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>bkw9</td>
<td>100</td>
<td>50</td>
<td>150</td>
</tr>
<tr>
<td>bkw10</td>
<td>200</td>
<td>70</td>
<td>150</td>
</tr>
<tr>
<td>bkw11</td>
<td>200</td>
<td>70</td>
<td>150</td>
</tr>
<tr>
<td>bkw12</td>
<td>500</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>bkw13</td>
<td>3152</td>
<td>640</td>
<td>960</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>177.69</td>
</tr>
</tbody>
</table>
Table 14  Test instances class comparison among the branch and bound procedures proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Name</th>
<th>New Exact Algorithm</th>
<th>Alvarez-Valdes et al. (2007)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td>W</td>
<td>L</td>
</tr>
<tr>
<td>Class 1</td>
<td>20</td>
<td>10</td>
<td>61.2</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>10</td>
<td>121.8</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>10</td>
<td>188.5</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>10</td>
<td>262.6</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>10</td>
<td>304.8</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>187.8</td>
</tr>
<tr>
<td>Class 2</td>
<td>20</td>
<td>30</td>
<td>19.7</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>30</td>
<td>39.1</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>30</td>
<td>60.1</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>30</td>
<td>83.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>100.5</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>60.5</td>
</tr>
<tr>
<td>Class 3</td>
<td>20</td>
<td>40</td>
<td>162.4</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>40</td>
<td>331.6</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>40</td>
<td>504.0</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>40</td>
<td>707.7</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>40</td>
<td>100.5</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>508.1</td>
</tr>
<tr>
<td>Class 4</td>
<td>20</td>
<td>100</td>
<td>61.9</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
<td>123.9</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
<td>193.0</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
<td>267.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>322.0</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>193.6</td>
</tr>
<tr>
<td>Class 5</td>
<td>20</td>
<td>100</td>
<td>532.6</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>100</td>
<td>1072.1</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>100</td>
<td>1637.8</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>100</td>
<td>2284.8</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
<td>2635.3</td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
<td>1632.5</td>
</tr>
</tbody>
</table>
## Table 15

Test instances class comparison among the branch and bound procedures proposed in this paper and in the literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>New Exact Algorithm</th>
<th>Alvarez-Valdes et al. (2007)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>n</td>
</tr>
<tr>
<td>Class 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References


