ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Computers & Operations Research ▮▮▮ (▮▮▮) ▮▮▮–▮▮▮

computers & operations research

www.elsevier.com/locate/cor

New reduction procedures and lower bounds for the two-dimensional bin packing problem with fixed orientation

Jacques Carlier*, François Clautiaux, Aziz Moukrim

HeuDiaSyC, CNRS UMR 6599, Université de Technologie de Compiègne, BP 20529 60205 Compiègne, France

Abstract

The two-dimensional bin-packing problem (2*BP*) consists of minimizing the number of identical rectangles used to pack a set of smaller rectangles. In this paper, we propose new lower bounds for 2*BP* in the discrete case. They are based on the total area of the items after application of *dual feasible functions* (DFF). We also propose the new concept of *data-dependent dual feasible functions* (DDFF), which can also be applied to a 2*BP* instance. We propose two families of Discrete DFF and DDFF and show that they lead to bounds which strictly dominate those obtained previously. We also introduce two new reduction procedures and report computational experiments on our lower bounds. Our bounds improve on the previous best results and close 22 additional instances of a well-known established benchmark derived from literature.

1. Introduction

The two-dimensional bin-packing problem (2*BP*) consists of minimizing the number of identical rectangles used to pack a set of smaller rectangles. This problem can occur in industry if pieces of steel, wood, or paper have to be cut from larger rectangles. It can also be used to model the layout of a newspaper. It is *NP-hard* as it generalizes the classical one-dimensional bin-packing problem (1*BP*) [1].

* Corresponding author. Tel.: +33 2 44234489; fax: +33 2 44234477.

E-mail addresses: jacques.carlier@hds.utc.fr (J. Carlier), francois.clautiaux@hds.utc.fr (F. Clautiaux), aziz.moukrim@hds.utc.fr (A. Moukrim).

0305-0548/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.cor.2005.08.012

A 1*BP* instance D_1 is a pair (A, C). $A = \{a_1, \ldots, a_n\}$ is the set of items to pack and $C \in \mathbb{N}$ is the size of the bins. An item a_i has a size $c_i \in \mathbb{N}$. A 2*BP* instance D_2 is a pair (A, B). A is the set of items a_i to pack. An item a_i has a width w_i and a height h_i ($w_i, h_i \in \mathbb{N}$). We consider the version of the problem in which the items cannot be rotated. The position of item a_i, denoted by (x_i, y_i), corresponds to the coordinates of its bottom left-hand corner. The bin $B = (W, H)$ is of width W and height H. $OPT(D)$ denotes the minimum number of bins required for a given instance D.

In this paper, we describe two new reduction procedures which are used to reduce the size of the instance. They can be applied as a preprocessing step before applying a heuristic or an exact method. For this purpose, we introduce *identically-feasible functions* (IFF) which can be used to modify the size of the items without changing the value of *OPT*. Thus, we show that the initial instance can be reduced by removing small items and by increasing the size of the large items.

We also deal with lower bounds. The most recent bounds were introduced by Boschetti and Mingozzi [2]: they dominate the previous best bounds [3,4]. For 1*BP*, Haouari and Gharbi [5] have proposed a method to improve the value of a lower bound. They plan to generalize their method to 2*BP*. Several bounds are equal to the total area of the items for an instance obtained after applying a suitable function to the size of the items. This processing can be done using *dual-feasible functions* (DFF), introduced by Johnson [6] and used by Lueker [7] and Fekete and Schepers [8] for computing lower bounds for 1*BP*. Fekete and Schepers [3] have shown that these functions can also be applied to both dimensions of a 2*BP* instance to obtain lower bounds. The same concept is used by Carlier and Néron [9] for cumulative scheduling problems. The functions they use are a subset of the DFF, restricted to the discrete case. We will refer to this concept as *discrete dual-feasible functions* (Discrete DFF): the only difference is that we consider discrete values and that they are not defined from $[0, 1]$ to $[0, 1]$ but from $[0, X]$ to $[0, X']$ for X and X' integers. Caprara et al. [10] have also dealt with DFF. They propose an exact method to find the pair of DFF which leads to the best continuous bound for a given instance.

We use the concept of DFF and a new type of functions, the *data-dependent dual feasible functions* (DDFF), which are designed for a specific type of instances. These functions are used in the general framework proposed by Fekete and Schepers [3] for computing bounds for *multi-dimensional* packing problems. We also describe three families of functions (two families of DFF and a family of DDFF) and show that they lead to bounds which strictly dominate those proposed by Labbé et al. [11] and Martello and Vigo [4] for 1*BP* and Boschetti and Mingozzi [2] for 2*BP*. To our knowledge, our bounds are the best polynomial time computable bounds for 2*BP*.

We report computational results for the reduction procedures and for our lower bounds. We tested our methods on well-known benchmarks derived from literature [4,12] so we can compare our methods with the best known results [13] and to the method of Fekete and Schepers [3]. Our reduction procedures are very efficient for several classes of instances, as a large number of items are removed for many benchmarks. The results confirm that our method dominates the others for all instances and leads to improved results.

Section 2 is an overview of the methods used in the literature for reduction procedures and lower bounds for 1*BP* and 2*BP*, which are used in this article. In Section 3, we introduce the concept of IFF and we propose two new reduction procedures which use IFF. Section 4 is devoted to our new lower bounds. We show in Section 5 that they dominate previous lower bounds [2,4,11]. In Section 6, we compare our reduction procedures and our lower bounds to those proposed in literature. The results are strictly better for several instances, and allow us to close 22 additional instances.

2. Literature review

2.1. Reduction procedures

The following reduction procedure is proposed by Boschetti and Mingozzi [2]. For a given item a_j, one can compute the maximum width W_j^* less than W which can be reached by packing a set of items including a_j one above the other. The width of item a_j can be modified as follows:

$$w_j \leftarrow w_j + (W - W_j^*).$$

The value W_j^* is the optimal value of the following subset-sum problem:

$$W_j^* = w_j + \text{Max}\left\{ \sum_{a_i \in A - \{a_j\}} w_i \xi_i : \sum_{a_i \in A - \{a_j\}} w_i \xi_i \leqslant W - w_j, \xi_i \in \{0, 1\} \right\}.$$

A classical pseudo-polynomial algorithm can compute W_j^* in $O(Wn)$ time. The reduction procedure depends on the order in which the items a_j are considered. A heuristic criterion can be used to choose this order. For example, it is interesting to increase the width of items which have a large height because the total area obtained is larger. A similar procedure can be obtained by exchanging widths with heights.

2.2. Lower bounds

In this section, we recall several important results concerning the so-called DFF and describe previous lower bounds for both 1*BP* and 2*BP*.

2.2.1. Dual feasible functions

For 2*BP* (resp., 1*BP*), a well-known bound is the continuous bound L_0. The idea is to calculate the total area (resp., size) of the items and to divide the obtained value by the area (resp., size) of the bin. The rounded up value obtained is a valid lower bound. For 2*BP*, $L_0 = \lceil \sum_{a_i \in A} w_i h_i / WH \rceil$.

This bound does not take into account the fact that many items cannot be packed together in a bin. One way of improving the continuous bound is to modify the size of the items so that the obtained instance is feasible if the initial one is. It can be done using DFF, introduced by Johnson [6].

Definition 2.1. $g : [0, 1] \to [0, 1]$ is a DFF if for any finite set S of real numbers, we have

$$\sum_{x \in S} x \leqslant 1 \quad \Rightarrow \quad \sum_{x \in S} g(x) \leqslant 1.$$

Using DFF on a 1*BP* instance is immediate, and several papers propose lower bounds for this problem which use DFF [7,8]. Fekete and Schepers [3] have shown that DFF can be applied to both dimensions of a 2*BP* instance to obtain valid lower bounds. The authors also propose three DFF and show that the bounds obtained dominate the bounds of Martello and Vigo [4]. We denote their bound $L_{\text{FS}}^{\text{1D}}$ for 1*BP* and $L_{\text{FS}}^{\text{2D}}$ for 2*BP*.

Carlier and Néron [9,14] use the similar concept of *Redundant Function* in order to get lower bounds for scheduling problems. As this concept is close to the DFF, we will refer to these functions as Discrete DFF.

Definition 2.2. A Redundant Function, or *Discrete DFF f* is a discrete application from $[0, X]$ to $[0, X']$ (X and X' integers) such that

$$x_1 + x_2 + \cdots + x_k \leqslant X \quad \Rightarrow \quad f(x_1) + f(x_2) + \cdots + f(x_k) \leqslant f(X) = X'.$$

Note that for every Discrete DFF, there is an equivalent DFF defined in $[0, 1]$. We use this notation to fit in with the notation of Carlier and Néron [9], and to simplify the formulation of the bounds.

2.2.2. Lower bound L^{1D}_{MV} (Martello and Vigo [4])

Martello and Vigo [4] propose a bound L^{1D}_{MV} for 1*BP*. First, they introduce a decomposition of the set A of items into three subsets depending on a given parameter k: the *large* items (A_{large}), the *medium* ones (A_{med}), and the *small* ones (A_s). Let k be an integer, $1 \leqslant k \leqslant \frac{1}{2}C$, $A_{large} = \{a_i \in A : c_i > C - k\}$, $A_{med} = \{a_i \in A : C - k \geqslant c_i > \frac{1}{2}C\}$ and $A_s = \{a_i \in A : \frac{1}{2}C \geqslant c_i \geqslant k\}$. The smallest items are not considered. L^{1D}_{MV} is computed as the maximum of the two lower-bounds L_α and L_β introduced below.

The bound L_α is based on the observation that at least as many bins as the number of large and medium items are needed to pack A. If there are too few medium items, this bound is equal to the sum of the number of large items and the continuous bound for the remaining items.

$$L_\alpha = \max_{1 \leqslant k \leqslant (1/2)C} \left\{ |A_{large} \cup A_{med}| + \max \left\{ 0, \left\lceil \frac{\sum_{a_i \in A_{med} \cup A_s} c_i}{C} - |A_{med}| \right\rceil \right\} \right\}.$$

The bound L_β estimates the number of small items which cannot be packed together with the medium ones and then evaluates how many bins are needed to pack them:

$$L_\beta = \max_{1 \leqslant k \leqslant (1/2)C} \left\{ |A_{large} \cup A_{med}| + \max \left\{ 0, \left\lceil \frac{|A_s| - \sum_{a_i \in A_{med}} \lfloor (C - c_i)/k \rfloor}{\lfloor C/k \rfloor} \right\rceil \right\} \right\}.$$

2.2.3. Lower bound L^{1D}_{Lab} (Labbé et al. [11])

The following lower bound L^{1D}_{Lab} is proposed by Labbé et al. [11] and analyzed by Bourjolly and Rebetez [15]. Bourjolly and Rebetez [15] show that it dominates the bound proposed by Martello and Toth [16].

The formulation we give for this bound is slightly different from the original formulation [11] but better fits our notation. For a given parameter $k \in [0, C/3]$, the bound proposed is computed as follows: let $A_{large} = \{a_i \in A : c_i > C - k\}$ be the set of large items, $A_{med} = \{a_i \in A : C/2 < c_i \leqslant C - k\}$ be the set of *medium* items, $A_d = \{a_i \in A : C/3 < c_i \leqslant C/2\}$ be the set of *small* items, and A^+_d the set of items of A_d which cannot be packed in the same bin as any item of A_{med}. We also introduce the following set: $A' = \{a_i \in A : k \leqslant c_i \leqslant C - k\}$.

$$L^{1D}_{Lab} = \max_{0 \leqslant k \leqslant C/3} \left\{ |A_{large}| + \max \left\{ |A_{med}| + \left\lceil \frac{|A^+_d|}{2} \right\rceil, \left\lceil \frac{\sum_{a_i \in A'} c_i}{C} \right\rceil \right\} \right\}.$$

2.2.4. Lower bound $L^{2D}_{1,BM}$ (Boschetti and Mingozzi [2])

Boschetti and Mingozzi [2] have proposed a lower bound $L^{2D}_{1,BM}$ which transforms the 2*BP* instance into a 1*BP* instance. We do not describe this bound as it is dominated by $L^{2D}_{2,BM}$ below.

2.2.5. Lower bound $L^{2D}_{2,BM}$ (Boschetti and Mingozzi [2])

Boschetti and Mingozzi [2] have proposed a bound which uses a lower bound for 1*BP* as subroutine. Let (p, q) be a pair of integers such that $1 \leqslant q \leqslant \frac{1}{2} W$ and $1 \leqslant p \leqslant \frac{1}{2} H$. We denote $A_{large} = \{a_i \in A : h_i > H - p \text{ and } w_i > W - q\}$, $A_{tall} = \{a_i \in A \backslash A_{large} : h_i > H - p \text{ and } w_i \geqslant q\}$, $A_{wide} = \{a_i \in A \backslash A_{large} : h_i \geqslant p \text{ and } w_i > W - q\}$, $A_s = \{a_i \in A \backslash A_{large} \cup A_{tall} \cup A_{wide} : h_i \geqslant p \text{ and } w_i \geqslant q\}$.

Boschetti and Mingozzi [2] propose a lower bound denoted $L^{2D}_{2,BM}$ for 2*BP* which can be split into two bounds. The first is based on a transformation of the problem into a 1*BP*. The second makes use of the fact that two *large* items of A_{large} cannot be packed in the same bin and that *tall* items of A_{tall} cannot be packed with *wide* items of A_{wide}.

$$L^{2D}_{2,BM} = \max_{1 \leqslant p \leqslant (1/2)H, 1 \leqslant q \leqslant (1/2)W} \{|A_{large}| + \max\{L'_2(p, q), L''_2(p, q)\}\},$$

where the bounds $L'_2(p, q)$ and $L''_2(p, q)$ are two evaluations for the minimum number of bins needed to pack the items of A_{tall}, A_{wide}, and A_s.

- In $L'_2(p, q)$, a 1*BP* instance D_{1BP} is created with a size of bin $C = WH$ as follows: for each item a_i in $A_{tall} \cup A_{wide} \cup A_s$ an item is created with size c_i. $c_i = w_i H$ if $a_i \in A_{tall}$, $c_i = W h_i$ if $a_i \in A_{wide}$, and $c_i = w_i h_i$ if $a_i \in A_s$. $L'_2(p, q)$ is obtained by computing a lower bound for D_{1BP} using L^{1D}_{MV} [4].
- In $L''_2(p, q)$ items of A_s are not considered. It makes use of the fact that an item of A_{tall} cannot be packed with an item of A_{wide}. These two sets are related to two-independent sub-problems. Moreover, neither two tall items nor two wide items can be packed side by side. Consequently, both problems can be seen as 1*BP* and lower bounds can be computed using L^{1D}_{MV}. Let $L^H_2(p, q)$ be a lower bound for the 1*BP* related to A_{tall} and $L^W_2(p, q)$ be a lower bound for the 1*BP* related to A_{wide}. $L''_2(p, q) = L^H_2(p, q) + L^W_2(p, q)$.

Theorem 2.1 (*Boschetti and Mingozzi [2]*). *$L^{2D}_{2,BM}$ is a valid lower bound for 2BP.*

2.2.6. Lower bound $L^{2D}_{3,BM}$ (Boschetti and Mingozzi [2])

$L^{2D}_{3,BM}$ uses a decomposition of the set of items into three sets depending on two parameters p and q: the big items (A_{large}), the medium items (A_{med}), and the small items (A_s). The method exploits the fact that the number of big and medium items is a lower-bound for 2*BP*. To improve this lower-bound $L^{2D}_{3,BM}$ evaluates the number of small items which can be packed together with the medium items. Then it computes an approximation for the number of bins needed to pack the remaining ones.

Let p and q be two integers such that $1 \leqslant p \leqslant \frac{1}{2} H$ and $1 \leqslant q \leqslant \frac{1}{2} W$, $A_{large} = \{a_i \in A : h_i > H - p \text{ and } w_i > W - q\}$, and $A_{med} = \{a_i \in A \backslash A_{large} : h_i > \frac{1}{2} H \text{ and } w_i > \frac{1}{2} W\}$, and $A_s = \{a_i \in A \backslash A_{large} \cup A_{med} : h_i \geqslant p \text{ and } w_i \geqslant q\}$. Several values have to be computed:

- $M_W(W, A_s)$: the maximum number of items of A_s which can be packed horizontally in a bin.
- $M_H(H, A_s)$: the maximum number of items of A_s which can be packed vertically in a bin.

- $M_W(W - w_i, A_s)$: the maximum number of items of A_s which can be packed horizontally with item a_i of A_{med} in a bin.
- $M_H(H - h_i, A_s)$: the maximum number of items of A_s which can be packed vertically with item a_i of A_{med} in a bin.

Each value is solution of a knapsack problem ($1KP$) defined in Section 4.

$$M_W(w^*, A_s) = \text{Max}\left\{\sum_{a_i \in A_s} \xi_i : \sum_{a_i \in A_s} w_i \xi_i \leqslant w^*, \xi_i \in \{0, 1\}\right\},$$

$$M_H(h^*, A_s) = \text{Max}\left\{\sum_{a_i \in A_s} \xi_i : \sum_{a_i \in A_s} h_i \xi_i \leqslant h^*, \xi_i \in \{0, 1\}\right\},$$

$$L^{2D}_{3,BM} = \max_{1 \leqslant p \leqslant (1/2)H, 1 \leqslant q \leqslant (1/2)W} \left\{|A_{large} \cup A_{med}| + \max\left\{0, \left\lceil \frac{|A_s| - \sum_{a_i \in A_{med}} m'(a_i, A_s)}{M_W(W, A_s) M_H(H, A_s)} \right\rceil\right\}\right\},$$

where $m'(a_i, A_s)$ is an upper bound of the number of items from A_s which can be packed together with a_i: $m'(a_i, A_s) = M_W(W, A_s) M_H(H - h_i, A_s) + M_W(W - w_i, A_s) M_H(H, A_s) - M_W(W - w_j, A_s) M_H(H - h_i, A_s)$.

Theorem 2.2 (*[Boschetti and Mingozzi [2]*). *$L^{2D}_{3,BM}$ is a lower bound for 2BP.*

2.2.7. Lower bound $L^{2D}_{4,BM}$ (Boschetti and Mingozzi [2])

The bound $L^{2D}_{4,BM}$ uses another method to evaluate the number of small items which cannot be packed with the large items and the number of bins needed to pack them. Each item a_i of small width (resp., height) has its width (resp., its height) reduced to a value computed from its size and a given parameter q (resp., p). The size of any large item a_i is increased so that the number of small items which can be packed together with a_i is the same. Let $A_{large} = \{a_i \in A : h_i > H - p \text{ and } w_i > W - q\}$, $A_{med} = \{a_i \in A \backslash A_{large} : h_i > \frac{1}{2}H \text{ and } w_i > \frac{1}{2}W\}$, $A^t_s = \{a_i \in A : h_i > \frac{1}{2}H \text{ and } \frac{1}{2}W \geqslant w_i \geqslant q\}$, $A^w_s = \{a_i \in A : \frac{1}{2}H \geqslant h_i \geqslant p \text{ and } w_i > \frac{1}{2}W\}$, $A^s_s = \{a_i \in A : \frac{1}{2}H \geqslant h_i \geqslant p \text{ and } \frac{1}{2}W \geqslant w_i \geqslant q\}$.

$$L^{2D}_{4,BM} = \max_{1 \leqslant p \leqslant H/2, 1 \leqslant q \leqslant W/2} \left\{|A_{large} \cup A_{med}| + \max\left\{0, \left\lceil \frac{\sum_{a_i \in A \backslash A_{large}} m''(a_i, p, q)}{\lfloor H/p \rfloor \lfloor W/q \rfloor} \right\rceil\right\}\right\},$$

where

- $m''(a_i, p, q) = \lfloor (W - w_i)/q \rfloor \lfloor (H - h_i)/p \rfloor - \lfloor (W - w_i)/q \rfloor \lfloor H/p \rfloor - \lfloor W/q \rfloor \lfloor (H - h_i)/p \rfloor$, if $a_i \in A_{med}$.

- $m''(a_i, p, q) = \lfloor w_i/q \rfloor \lfloor H/p \rfloor - \lfloor w_i/q \rfloor \lfloor (H - h_i)/p \rfloor$, if $a_i \in A_s^t$.
- $m''(a_i, p, q) = \lfloor W/q \rfloor \lfloor h_i/p \rfloor - \lfloor (W - w_i)/q \rfloor \lfloor h_i/p \rfloor$, if $a_i \in A_s^w$.
- $m''(a_i, p, q) = \lfloor w_i/q \rfloor \lfloor h_i/p \rfloor$, if $a_i \in A_s^s$.

Theorem 2.3 (*[Boschetti and Mingozzi [2]*). *$L_{4,\mathrm{BM}}^{2\mathrm{D}}$ is a lower bound for 2BP.*

The overall lower bound proposed by Boschetti and Mingozzi is the maximum among $L_{2,\mathrm{BM}}^{2\mathrm{D}}$, $L_{3,\mathrm{BM}}^{2\mathrm{D}}$, and $L_{4,\mathrm{BM}}^{2\mathrm{D}}$. We denote this bound $L_{\mathrm{BM}}^{2\mathrm{D}}$.

To sum up, Fekete and Schepers [3] have proposed a general framework to use DFF to compute lower bounds for 2*BP*. Another approach is used by Boschetti and Mingozzi [2], which leads to better bounds. We show in the following of the paper that the two approaches can be gathered and improved to obtain better results.

3. Reduction procedures: identically-feasible functions

In this section, we describe cases where some items could be removed in a 2*BP*. In the general case, if an item is too small, it may have no impact on the value *OPT*, and if it is of size (W, H) it forms a trivial sub-instance which can be separately treated. There are few such large items in the initial instance but their number can be increased by applying an efficient reduction procedure. This section deals with methods which increase the size of the large items and remove the small items which have no impact on *OPT*. The number of removed items can be used to compare reduction procedures, as well as the new total area obtained. It is useful to apply this sort of procedure as a preprocessing before using a heuristic or an exact method.

For this purpose, we define the concept of IFF. We show that the method proposed by Boschetti and Mingozzi [2] is equivalent to applying an IFF. Then we propose two new functions which remove small items and increase the size of the large items.

Definition 3.1. Let $D = (A, B)$ be a 2*BP* instance. A function f is an IFF associated with instance D iff instance $f(D) = (\{a'_1, \ldots, a'_n\}, B)$ obtained by applying f to all items of A is such that $OPT(D) = OPT(f(D))$.

The following proposition shows that the preprocessing proposed by Boschetti and Mingozzi [2] (see Section 2) can be obtained by applying an IFF. We denote this function u_w^j.

Proposition 3.1 (*Boschetti and Mingozzi [2]*). *Let D be an instance of 2BP and $a_j \in A$ an item.*

$$u_w^j : A \to A^1,$$
$$a_i \mapsto a'_i \quad \text{with} \quad \begin{cases} w'_i = w_i + (W - W_j^*) \text{ and } h'_i = h_i & \text{if } i = j, \\ w'_i = w_i \text{ and } h'_i = h_i & \text{otherwise} \end{cases}$$

is an IFF associated with instance D.

Similarly, the function u_h^j is defined by exchanging widths with heights. We denote with u the iterative application of u_h^j and u_w^j using a greedy criterion for the choice of a_j.

3.1. New identically-feasible function v_1

If the small items are sufficiently small, they can be packed into the available area around the large items. The previous method does not take this fact into account. The two functions we propose in this paper are designed to detect such cases. Consider a set A_{tall} of tall items, and a set A_{shallow} of items which can be packed above an item of A_{tall}. If all items of A_{shallow} can be packed into the area above the items of A_{tall}, these small items have no impact on the value of *OPT*.

Let D be an instance of 2*BP* and $1 \leqslant p \leqslant \frac{1}{2}H$ an integer. We define $A_{\text{tall}} = \{a_i \in A : h_i > H - p\}$ and $A_{\text{shallow}} = \{a_i \in A : h_i < p\}$. We denote $m = |A_{\text{tall}}|$. For each item a_i of A_{tall}, a bin B_i of size $(H - h_i, w_i)$ is created. Consider $D' = (A_{\text{shallow}}, \{B_1, \ldots, B_m\})$ the following decision problem: "*Is there a packing for the items of* A_{shallow} *in the bins* $B_1, \ldots, B_m$?".

Theorem 3.1. *If D' has a feasible solution then function*

$$v_1^p : A \to A^1,$$
$$a_i \mapsto a_i' \quad \text{where} \quad \begin{cases} w_i' = w_i \text{ and } h_i' = H & \text{if } h_i > H - p, \\ w_i' = 0 \text{ and } h_i' = 0 & \text{if } h_i < p, \\ w_i' = w_i \text{ and } h_i' = h_i & \text{otherwise} \end{cases}$$

is an IFF associated with instance D.

Proof. Consider the initial instance D and the modified instance $v_1^p(D)$ obtained by applying IFF v_1^p to D. Suppose an optimal packing is known for $v_1^p(D)$. From this packing an optimal packing for D can be built as follows: items in $A - A_{\text{shallow}}$ are packed in the same way, and the items in A_{shallow} are packed into the area available above the tall items (this area is available, as no items in $A - A_{\text{shallow}}$ can be packed above an item of A_{tall}). Using a similar method an optimal packing for $v_1^p(D)$ can be built from any optimal packing for D. If in the optimal packing obtained for D the items of A_{shallow} are not packed above items of A_{tall}, a similar packing can be built by packing the small items above the tall items (if necessary shifting the tall items up or down). From the new packing obtained a packing for $v_1^p(D)$ can be built using the same coordinates for the remaining items. □

We denote v_1^H the iterative application of v_1^p for the different values of p. This algorithm can be implemented in $O(H \times \alpha(n))$, where $\alpha(n)$ is the complexity of the algorithm used to solve the decision problem D'. If we take into account the fact that the only interesting values for p are those equal to the height of an item, the complexity becomes $O(n\alpha(n))$.

We define a similar function v_1^W by exchanging heights with widths. When the size of an item is increased in one dimension, decision problems which had no solutions before can become feasible. We denote v_1 the iterative application of v_1^W and v_1^H. The method terminates when the instance has not been modified by the previous step. The decision problems to solve are NP-complete but a heuristic is sufficient to obtain good results, as shown in the computational experiments (Section 6). Function v_1 is illustrated by Fig. 1.

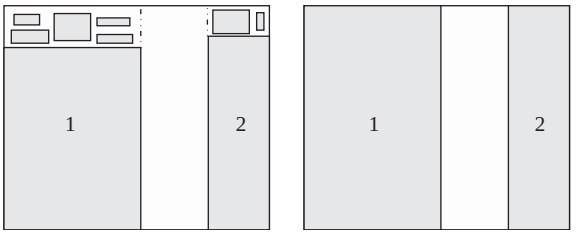

Fig. 1. Function v_1.

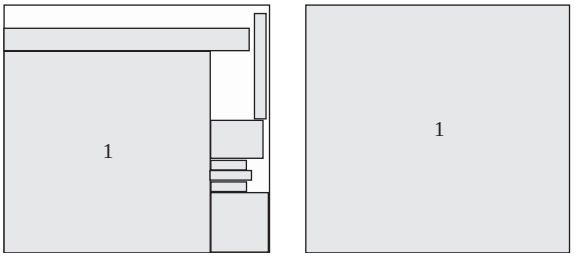

Fig. 2. Function v_2.

3.2. New identically-feasible function v_2

The reduction procedure related to function v_2 takes into account the whole area around the large items. Consider a set A_{large} of items of size greater than half of the bin in the two dimensions, and a set A_s of items which can be packed with an item of A_{large}. If all items in A_s can be packed with items of A_{large} these items have no impact on the value of *OPT* (see Fig. 2).

Let D be an instance of 2*BP* and $1 \leqslant p \leqslant \frac{1}{2} H$ and $1 \leqslant q \leqslant \frac{1}{2} W$ two integers. We denote A_{large} the set of items a_i such that $w_i > W - q$ and $h_i > H - p$ and A_s the set of items a_i such that $w_i < q$ or $h_i < p$. Let D' be the following 2*BP*: $(A' = A_s \cup A_{\text{large}}, B)$.

Theorem 3.2. *If* $OPT(D') = |A_{\text{large}}|$, *then function*

$$v_2^{p,q} : A \to A^1,$$

$$a_i \mapsto a_i' \quad \text{where} \quad \begin{cases} w_i' = W \text{ and } h_i' = H & \text{if } w_i > W - q \text{ and } h_i > H - p, \\ w_i' = 0 \text{ and } h_i' = 0 & \text{if } w_i < q \text{ or } h_i < p, \\ w_i' = w_i \text{ and } h_i' = h_i & \text{otherwise} \end{cases}$$

is an IFF associated with instance D.

Proof. Consider the initial instance D and the modified instance $v_2^{p,q}(D)$. Suppose a packing is known for $v_2^{p,q}(D)$. From this packing, a packing for D can be built as follows: items of $A - A_s$ are packed in the same way, and the items of A_s are packed in the area available around the large items (this area is available as no items of $A - A_s$ can be packed with an item of A_{large}). Using a similar method, a packing for $v_2^{p,q}(D)$ can be obtained from D. If in the packing obtained for D, the items of A_s are not packed

with an item of A_{large}, a similar packing can be built by packing the small items in the same bins as the large items (if necessary shifting the large items up or down, or from left to right). From the new packing obtained, a packing for $v_2^{p,q}(D)$ can be built using the same coordinates for the remaining items. □

The optimal value for the generated 2*BP* instance D' is difficult to compute, but a simple heuristic can detect a large number of trivial cases. We denote v_2 the iterative application of $v_2^{p,q}$ for the different values of p and q. This algorithm can be implemented in $O(HW\alpha(n))$, where $\alpha(n)$ is the complexity of the heuristic used to find a solution for the generated 2*BP* instance. This complexity can be reduced using the following observation: the only interesting values for p and q are those which are equal to the size of items of A. Consequently, the complexity of the method is $O(n^2\alpha(n))$.

4. New lower bounds

In the literature, DFF have been successfully applied to obtain lower bounds for 2*BP*. It appears that there are bounds which cannot be computed by the mean of DFF. So, we propose a new concept of functions which are dependent on the data, and lead to improving bounds.

4.1. Data-dependent dual-feasible functions

DFF are defined independently of the instance. Another class of functions can be defined which depends on the sizes of items in the instance. Once again, we deal with a class of discrete functions.

Definition 4.1. Let $I = \{1, \ldots, n\}$, $c_1, c_2, \ldots, c_n$ n integer values and C an integer such that $C \geqslant c_i$ for $i = 1, \ldots, n$. A DDFF f associated with C and $c_1, c_2, \ldots, c_n$ is a discrete application from $[0, C]$ to $[0, C']$ such that

$$\forall I_1 \subset I, \quad \sum_{i \in I_1} c_i \leqslant C \;\Rightarrow\; \sum_{i \in I_1} f(c_i) \leqslant f(C) = C'.$$

For a given instance D, a function dependent on instance D can be used as if it were a Discrete DFF. This family of functions takes into account the number of occurrences of each value, and thus can lead to bounds which strictly dominate those obtained by the DFF. We now give the example of a DDFF g which is better than any DFF for a specific case. Let 5, 6, 7 be the list of values and $C = 10$. $g(5) = 10$, $g(6) = 10$, $g(7) = 10$, $g(10) = 10$. g is not a Discrete DFF since $20 = g(5) + g(5) > g(10) = 10$.

Theorem 4.1. *Let D be a 2BP instance and D' be the instance obtained after applying f, a DFF or a DDFF on the width* (*resp., height*) *of D. Any lower bound for D' is also a lower bound for D.*

Proof. Clearly, any feasible packing for D remains feasible for D' (see [3]). So the set of possible packings associated with D' contains the set of solutions associated with D. □

4.2. New Discrete DFF and DDFF

We shall now describe three families of functions f_0^k, f_1^k and f_2^k. These families are inspired from the bounds proposed by Boschetti and Mingozzi [2], but they can be applied to each dimension separately. f_0^k and f_2^k are Discrete DFF, whereas f_1^k is a DDFF. We show in the next section that families f_0^k, f_1^k and f_2^k can be combined to obtain bounds which strictly dominate the bounds of Boschetti and Mingozzi [2]. In the following, C denotes an integer value and k an integer smaller than $\frac{C}{2}$. As stated in Section 2, we use a discrete version of DFF.

4.2.1. Classical function f_0^k

We introduce a classical DFF defined for a set of integers. It is based on a simple observation: if no items can be packed with an item a_i, the size of a_i can be increased to the size of the bin. Analogously, given a value k ($1 \leqslant k \leqslant C/2$), the size of the items of size strictly greater than $C-k$ can be increased if any item smaller than k is removed. In this paper f_0^k denotes the corresponding family of DFF. This function has also been used by Fekete and Schepers [3]. They also have shown that this function is underlying bounds described in [16,17].

$$f_0^k : [0, C] \to [0, C],$$
$$x \mapsto \begin{cases} C & \text{if } x > C-k, \\ x & \text{if } C-k \geqslant x \geqslant k, \\ 0 & \text{otherwise.} \end{cases}$$

4.2.2. Function f_1^k

Let f_1^k be a function defined for a given parameter k, $1 \leqslant k \leqslant \frac{1}{2}C$ and a list of integer values $c_1, c_2, \ldots, c_n$ ($I = \{1, \ldots, n\}$). We introduce the set $J = \{i \in I : \frac{1}{2}C \geqslant c_i \geqslant k\}$ and $M_C(X, J)$ the optimal value for the one-dimensional $1KP$ related to J and size X. This value is equal to the maximum number of items c_i such that $i \in J$ which can be packed together in a container of size X. It can be solved in linear time if the items are sorted by increasing order of size. This function is underlying the bound $L_{3,\mathrm{BM}}^{2\mathrm{D}}$ of Boschetti and Mingozzi [2].

$$f_1^k : [0, C] \to [0, M_C(C, J)],$$
$$x \mapsto \begin{cases} M_C(C, J) - M_C(C-x, J) & \text{if } x > \frac{1}{2}C, \\ 1 & \text{if } \frac{1}{2}C \geqslant x \geqslant k, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 4.2. *Function f_1^k is a DDFF for the list $c_1, \ldots, c_n$ and the value C.*

Proof. Suppose f_1^k is not a DDFF. Consequently, there exists a subset I_1 of integers such that $\sum_{i \in I_1} c_i \leqslant C$ and $\sum_{i \in I_1} f_1^k(c_i) > M_C(C, J)$. We consider different cases depending on whether or not there exists $j \in I_1$ such that $c_j > \frac{1}{2}C$. Items of size less than k are not considered.

1. There exists $j \in I_1$ such that $c_j > \frac{1}{2}C$. Note that for $i \neq j, i \in J$. We have $f_1^k(c_j) + \sum_{i \in I_1 - \{j\}} f_1^k(c_i) > M_C(C, J)$, and thus $M_C(C, J) - M_C(C - c_j, J) + \sum_{i \in I_1 - \{j\}} 1 > M_C(C, J)$. Since $\sum_{i \in I_1} c_i \leqslant C$,

$M_C(C - c_j, J) \geqslant |I_1 - \{j\}|$. So, $\sum_{i \in I_1 - \{j\}} 1 = |I_1 - \{j\}| > M_C(C - c_j, J) \geqslant |I_1 - \{j\}|$. There is a contradiction.

2. For all $i \in I_1$, $\frac{1}{2} C \geqslant c_i \geqslant k$ ($I_1 \subseteq J$). In this case, $M_C(C, J) \geqslant |I_1|$. So, we have $|I_1| = \sum_{i \in I_1} 1 > M_C(C, J) \geqslant |I_1|$. There is a contradiction. □

Note that f_1^k is not a DFF as $M_C(C, J)$ is dependent on the values $c_1, \ldots, c_n$.

4.2.3. Function f_2^k

Function f_2^k is based on a special rounding technique. It is an improvement on a function which has been indirectly used by Boschetti and Mingozzi [2] and as a DFF by Boschetti [18].

Let f_2^k be the function defined as follows:

$$f_2^k : [0, C] \to \left[0, 2\left\lfloor \frac{C}{k} \right\rfloor\right],$$

$$x \mapsto \begin{cases} 2\left(\left\lfloor \dfrac{C}{k} \right\rfloor - \left\lfloor \dfrac{C - x}{k} \right\rfloor\right) & \text{if } x > \frac{1}{2} C, \\ \left\lfloor \dfrac{C}{k} \right\rfloor & \text{if } x = \frac{1}{2} C, \\ 2\left\lfloor \dfrac{x}{k} \right\rfloor & \text{if } \frac{1}{2} C > x. \end{cases}$$

Theorem 4.3. *Function f_2^k is a DFF.*

Proof. Suppose function f_2^k is not a DFF. Then there exists a list of integers $c_1, \ldots, c_n$ ($I = \{1, \ldots, n\}$) such that $\sum_{i \in I} c_i \leqslant C$ and $\sum_{i \in I} f_2^k(c_i) > 2\lfloor C/k \rfloor$.

The cases to be considered depend on the values in the list.

1. There is a value $j \in I$ such that $c_j > \frac{1}{2} C$. We have $2\lfloor C/k \rfloor - 2\lfloor (C - c_j)/k \rfloor + 2 \sum_{i \in I - \{j\}} \lfloor c_i/k \rfloor > 2\lfloor C/k \rfloor$. Thus,

$$\sum_{i \in I - \{j\}} \left\lfloor \frac{c_i}{k} \right\rfloor > \left\lfloor \frac{C - c_j}{k} \right\rfloor. \tag{1}$$

Recall that $\sum_{i \in I - \{j\}} c_i \leqslant C - c_j$. Then,

$$\sum_{i \in I - \{j\}} \left\lfloor \frac{c_i}{k} \right\rfloor \leqslant \left\lfloor \sum_{i \in I - \{j\}} \frac{c_i}{k} \right\rfloor \leqslant \left\lfloor \frac{C - c_j}{k} \right\rfloor. \tag{2}$$

Eq. (2) contradicts with Eq. (1).

2. There are two values $j, l \in I$ such that $c_j = c_l = C/2$. The contradiction is immediate.
3. There is only one value $j \in I$ such that $c_j = C/2$. We have $\lfloor C/k \rfloor + 2 \sum_{i \in I - \{j\}} \lfloor c_i/k \rfloor > 2\lfloor C/k \rfloor$. Thus, $2 \sum_{i \in I - \{j\}} \lfloor \frac{c_i}{k} \rfloor > \lfloor C/k \rfloor$. Recall that $\sum_{i \in I - \{j\}} c_i \leqslant C/2$. Then, $2 \sum_{i \in I - \{j\}} c_i/k \leqslant C/k$. So

we obtain

$$\left\lfloor \frac{C}{k} \right\rfloor < 2 \sum_{i \in I - \{j\}} \left\lfloor \frac{c_i}{k} \right\rfloor \leqslant \left\lfloor 2 \sum_{i \in I - \{j\}} \frac{c_i}{k} \right\rfloor \leqslant \left\lfloor \frac{C}{k} \right\rfloor.$$

There is a contradiction.

4. For each $i \in I$, $c_i < \frac{1}{2}C$. We have $\sum_{i \in I} c_i \leqslant C$ and $2 \sum_{i \in I} \lfloor c_i/k \rfloor > 2 \lfloor C/k \rfloor$. So we have

$$\sum_{i \in I} \left\lfloor \frac{c_i}{k} \right\rfloor \leqslant \left\lfloor \frac{\sum_{i \in I} c_i}{k} \right\rfloor \leqslant \left\lfloor \frac{C}{k} \right\rfloor \quad \text{and} \quad \sum_{i \in I} \left\lfloor \frac{c_i}{k} \right\rfloor > \left\lfloor \frac{C}{k} \right\rfloor.$$

There is a contradiction. □

4.3. Three new lower bounds for 2BP

4.3.1. Using DDFF

We first introduce bounds dedicated to 1*BP*. For $u = 0, 1, 2$ we define $L^{1BP}_{f_u}$:

$$L^{1BP}_{f_u} = \max_{1 \leqslant k \leqslant C/2} \left\{ \left\lceil \frac{\sum_{a_i \in A} f^k_u(c_i)}{f^k_u(C)} \right\rceil \right\},$$

$$L^{1D}_{CCM} = \max_{u \in \{0,1,2\}} \{L^{1BP}_{f_u}\}.$$

We introduce nine analogous bounds for 2*BP*, following the framework defined by Fekete and Schepers [3]. For $u = 0, 1, 2$ and $v = 0, 1, 2$ we define $L_{(f_u, f_v)}$:

$$L^{2D}_{(f_u, f_v)} = \max_{1 \leqslant k \leqslant W/2, 1 \leqslant l \leqslant H/2} \left\{ \left\lceil \frac{\sum_{a_i \in A} f^k_u(w_i) f^l_v(h_i)}{f^k_u(W) f^l_v(H)} \right\rceil \right\}.$$

Let $L^{2D}_{1,CCM}$ be the maximum continuous bound among the instances obtained after applying a DDFF on each dimension.

$$L^{2D}_{1,CCM} = \max_{u \in \{0,1,2\}, v \in \{0,1,2\}} \{L^{2D}_{(f_u, f_v)}\}.$$

4.3.2. Using the scheme of Boschetti and Mingozzi [2]

It appears that the lower-bound $L^{2D}_{2,BM}$ proposed by Boschetti and Mingozzi [2] cannot be computed by the mean of DFF. So we have to use the scheme underlying this bound to improve on the obtained results. In both bounds, a lower-bound L^{1D}_{MV} proposed in [4] for 1*BP* is used. Let $L^{2D}_{2,CCM}$ be the bound obtained from $L^{2D}_{2,BM}$ by replacing L^{1D}_{MV} by L^{1D}_{CCM}.

Proposition 4.1. *$L^{2D}_{2,CCM}$ is a valid lower bound for 2BP.*

Proof. Boschetti and Mingozzi [2] have shown that their bound $L^{2D}_{2,BM}$ is a valid lower bound for 2*BP*. The only difference between $L^{2D}_{2,BM}$ and $L^{2D}_{2,CCM}$ is the inner 1*BP* lower bound used. L^{1D}_{CCM} is valid, so $L^{2D}_{2,CCM}$ is also valid. □

We obtain the following valid lower bound: $L^{2D}_{CCM} = \max\{L^{2D}_{1,CCM}, L^{2D}_{2,CCM}\}$. This new lower bound improves previous results. Compared to the bounds $L^{2D}_{3,BM}$ and $L^{2D}_{4,BM}$ proposed in [2], $L^{2D}_{1,CCM}$ generalizes several methods. The height and the width are considered separately, which leads to new combinations. We have also proposed function f^k_2, which improves on the previously used function. $L^{2D}_{2,CCM}$ improves $L^{2D}_{2,BM}$ proposed by Boschetti and Mingozzi [2] by replacing lower-bound L^{1D}_{MV} by L^{1D}_{CCM}. Note that the two methods have the same complexity. We show in the computational experiments that we can implement our bounds in such a way that they are faster than the bounds of Boschetti and Mingozzi.

4.3.3. *Improving on L^{2D}_{CCM} by iteratively removing items*

When only DFF are used, removing an item can only decrease the value of the lower bound obtained. When DDFF are used, this observation does not hold any more. So a way of improving the lower bounds is to remove items and apply L^{2D}_{CCM} on the residual instance. The items are sorted following a given order. At each step of an iterative process, the last item is removed and the lower bound is computed. The algorithm stops when the number of items is less than or equal to the current best lower bound. This method may improve the results in some cases. We denote this method as $L^{2D}_{3,CCM}$.

5. Dominance relations

We now show that the bound L^{2D}_{CCM} dominates L^{2D}_{BM} proposed by Boschetti and Mingozzi [2]. We first show that L^{1D}_{CCM} dominates L^{1D}_{MV} proposed by Martello and Vigo [4] and L^{1D}_{Lab} proposed by Labbé et al. [11] for 1*BP*.

5.1. *Lower-bound L^{1D}_{MV} for 1BP [4]*

Proposition 5.1. *The bound L^{1D}_{CCM} dominates the bound L_α proposed by Martello and Vigo [4].*

Proof. For a given k, $L_\alpha(k) = |A_{large}| + \max\{|A_{med}|, \lceil \sum_{a_i \in A_{med} \cup A_s} c_i / C \rceil\}$. There are two possible cases:

1. If $\sum_{a_i \in A_{med} \cup A_s} c_i / C \leqslant |A_{med}|$, $L_\alpha(k) = |A_{large} \cup A_{med}|$. In this case, for each k, $L_\alpha(k)$ makes use of the fact that the number of large and medium items is a lower bound for 2*BP*. Now, consider the previously defined DFF $f^{(1/2)C}_0$. The continuous bound for the modified instance obtained is

$$L^{1D}_{f^{(1/2)C}_0} = \left\lceil \frac{\sum_{a_i \in A_{large} \cup A_{med}} C + \sum_{\{a_i \in A : c_i = (1/2)C\}} \frac{1}{2} C}{C} \right\rceil,$$

$$L^{1D}_{f^{(1/2)C}_0} = \left|A_{large} \cup A_{med}\right| + \tfrac{1}{2} \left|\{a_i \in A : c_i = \tfrac{1}{2} C\}\right|.$$

In this particular case, $L_\alpha(k)$ is dominated by L_{f_0} and thus by L^{1D}_{CCM}.

2. If

$$\frac{\sum_{a_i \in A_{\text{med}} \cup A_{\text{s}}} c_i}{C} > |A_{\text{med}}|, \quad L_\alpha(k) = |A_{\text{large}}| + \left\lceil \frac{\sum_{a_i \in A_{\text{med}} \cup A_{\text{s}}} c_i}{C} \right\rceil.$$

In this case, $L_\alpha(k)$ is equal to the continuous bound after application of DFF f_0^k. So it is dominated by $L_{\text{CCM}}^{\text{1D}}$. □

Proposition 5.2. *The bound $L_{\text{CCM}}^{\text{1D}}$ dominates the bound L_β proposed by Martello and Vigo [4].*

Proof. For a given k,

$$L_\beta(k) = |A_{\text{large}} \cup A_{\text{med}}| + \max\left\{0, \left\lceil \frac{|A_{\text{s}}| - \sum_{a_i \in A_{\text{med}}} \left\lfloor \frac{C - c_i}{k} \right\rfloor}{\left\lfloor \frac{C}{k} \right\rfloor} \right\rceil\right\}.$$

For the case where $L_\beta(k) = |A_{\text{large}} \cup A_{\text{med}}|$, see the proof of Proposition 5.1.
If

$$|A_{\text{s}}| > \sum_{a_i \in A_{\text{med}}} \left\lfloor \frac{C - c_i}{k} \right\rfloor, \quad L_\beta(k) = |A_{\text{large}} \cup A_{\text{med}}| + \left\lceil \frac{|A_{\text{s}}| - \sum_{a_i \in A_{\text{med}}} \left\lfloor \frac{C - c_i}{k} \right\rfloor}{\left\lfloor \frac{C}{k} \right\rfloor} \right\rceil.$$

The method used to evaluate how many items of A_{s} can be packed together in a bin involves reducing their size to a given value depending on parameter k. The size of each large item a_i is increased in such a way that the same number of small items can be packed with a_i. We show that a better bound can be obtained by applying the family of functions f_2^k. The continuous bound obtained for the instance after application of f_2^k is

$$L_{f_2^k}^{\text{1D}} = \left\lceil \frac{\sum_{a_i \in A_{\text{large}} \cup A_{\text{med}} \cup A_{\text{s}}} f_2^k(c_i)}{2\lfloor C/k \rfloor} \right\rceil.$$

To simplify the proof, we take into account the fact that f_2^k is an increasing function. Thus, we consider that $f_2^k(x) \geqslant 2\lfloor x/k \rfloor$ if $x = C/2$.

$$L_{f_2^k}^{\text{1D}} \geqslant \left\lceil \frac{2\sum_{a_i \in A_{\text{large}}} \left\lfloor \frac{C}{k} \right\rfloor + 2\sum_{a_i \in A_{\text{s}}} \left\lfloor \frac{c_i}{k} \right\rfloor + 2\sum_{a_i \in A_{\text{med}}} \left(\left\lfloor \frac{C}{k} \right\rfloor - \left\lfloor \frac{C - c_i}{k} \right\rfloor \right)}{2\left\lfloor \frac{C}{k} \right\rfloor} \right\rceil,$$

$$L_{f_2^k}^{\text{1D}} \geqslant |A_{\text{large}}| + |A_{\text{med}}| + \left\lceil \frac{\sum_{a_i \in A_{\text{s}}} \left\lfloor \frac{c_i}{k} \right\rfloor - \sum_{a_i \in A_{\text{med}}} \left\lfloor \frac{C - c_i}{k} \right\rfloor}{\left\lfloor \frac{C}{k} \right\rfloor} \right\rceil.$$

For $a_i \in A_s$, since $c_i \geqslant k$, $\lfloor c_i/k \rfloor \geqslant 1$.

$$L^{1D}_{f_2^k} \geqslant |A_{large}| + |A_{med}| + \left\lceil \frac{|A_s| - \sum_{a_i \in A_{med}} \left\lfloor \frac{C - c_i}{k} \right\rfloor}{\left\lfloor \frac{C}{k} \right\rfloor} \right\rceil = L_\beta(k),$$

$$L^{1D}_{CCM} \geqslant L^{1D}_{f_2^k} \geqslant L_\beta. \qquad \square$$

Since $L^{1D}_{MV} = \max\{L_\alpha, L_\beta\}$, and given Propositions 5.1 and 5.2, we obtain the following result.

Theorem 5.1. *The bound L^{1D}_{CCM} dominates L^{1D}_{MV} proposed by Martello and Vigo [4].*

5.2. Lower-bound L^{1D}_{Lab} for 1BP [11])

We now show that our bounds dominate the bound L^{1D}_{Lab} proposed by Labbé et al. [11].

Proposition 5.3. *L^{1D}_{CCM} dominates L^{1D}_{Lab}.*

Proof. For a given value of k,

$$L^{1D}_{Lab}(k) = |A_{large}| + \max\left\{ |A_{med}| + \left\lceil \frac{|A^+_d|}{2} \right\rceil, \left\lceil \frac{\sum_{a_i \in A'} c_i}{C} \right\rceil \right\}.$$

If

$$|A_{med}| + \left\lceil \frac{|A^+_d|}{2} \right\rceil \leqslant \left\lceil \frac{\sum_{a_i \in A'} c_i}{C} \right\rceil,$$

$L^{1D}_{Lab}(k)$ is equal to the continuous lower bound after applying function f^k_0.

If

$$|A_{med}| + \left\lceil \frac{|A^+_d|}{2} \right\rceil > \left\lceil \frac{\sum_{a_i \in A'} w_i}{C} \right\rceil, \quad L^{1D}_{Lab}(k) = |A_{large} \cup A_{med}| + \left\lceil \frac{|A^+_d|}{2} \right\rceil.$$

There exists a given parameter p ($p > C/3$) such that $c_i \geqslant p$ implies $a_i \in A^+_d$ and $c_i < p$ implies $a_i \notin A^+_d$. Now, let us compute the continuous bound $L^{1D}_{f_1^p}$ obtained after applying function f^p_1 to the instance.

$$L^{1D}_{f_1^p} = \left\lceil \frac{\sum_{a_i \in A} f^p_1(c_i)}{f^p_1(C)} \right\rceil.$$

Since $p \geqslant C/3$, so only items of $A_{\text{large}} \cup A_{\text{med}} \cup A_{\text{d}}^{+}$ have to be considered.

$$L_{f_1^p}^{1\text{D}} = \left\lceil \frac{\sum_{a_i \in A_{\text{large}} \cup A_{\text{med}}} f_1^p(c_i) + \sum_{a_i \in A_{\text{d}}^{+}} f_1^p(c_i)}{f_1^p(C)} \right\rceil,$$

$$L_{f_1^p}^{1\text{D}} = \left\lceil \frac{\sum_{a_i \in A_{\text{large}} \cup A_{\text{med}}} M_C(C, A_{\text{d}}^{+}) - M_C(C - c_i, A_{\text{d}}^{+}) + \sum_{a_i \in A_{\text{d}}^{+}} 1}{M_C(C, A_{\text{d}}^{+})} \right\rceil.$$

As no item of A_{d}^{+} can be packed with any item of $A_{\text{med}} \cup A_{\text{large}}$, $M_C(C - c_i, A_{\text{d}}^{+}) = 0$ for all items of $A_{\text{med}} \cup A_{\text{large}}$.

$$L_{f_1^p}^{1\text{D}} = \left\lceil \frac{\sum_{a_i \in A_{\text{large}} \cup A_{\text{med}}} M_C(C, A_{\text{d}}^{+}) + \sum_{a_i \in A_{\text{d}}^{+}} 1}{M_C(C, A_{\text{d}}^{+})} \right\rceil.$$

We have $p > C/3$, thus $M_C(C, A_{\text{d}}^{+}) \leqslant 2$, $L_{f_1^p}^{1\text{D}} \geqslant |A_{\text{large}} \cup A_{\text{med}}| + \lceil |A_{\text{d}}^{+}|/2 \rceil$. This completes the proof. □

5.3. Bounds $L_{1,\text{BM}}^{2\text{D}}$ and $L_{2,\text{BM}}^{2\text{D}}$ [2]

The only difference between $L_{2,\text{BM}}^{2\text{D}}$ and $L_{2,\text{CCM}}^{2\text{D}}$ is the inner lower bound for the 1*BP* created. From Theorem 5.1, $L_{\text{CCM}}^{1\text{D}}$ dominates $L_{\text{MV}}^{1\text{D}}$, so $L_{2,\text{CCM}}^{2\text{D}}$ dominates $L_{2,\text{BM}}^{2\text{D}}$. The following result holds.

Corollary 5.1. *The bound $L_{2,\text{CCM}}^{2\text{D}}$ dominates $L_{2,\text{BM}}^{2\text{D}}$ proposed in* [2].

Since Boschetti and Mingozzi [2] proved that $L_{1,\text{BM}}^{2\text{D}}$ is dominated by $L_{2,\text{BM}}^{2\text{D}}$, $L_{2,\text{CCM}}^{2\text{D}}$ dominates both.

5.4. The bound $L_{3,\text{BM}}^{2\text{D}}$ [2]

Theorem 5.2. *$L_{1,\text{CCM}}^{2\text{D}}$ dominates the bound $L_{3,\text{BM}}^{2\text{D}}$ proposed in* [2].

Proof.

$$L_{3,\text{BM}}^{2\text{D}}(p, q) = |A_{\text{large}} \cup A_{\text{med}}| + \max\left\{0, \left\lceil \frac{|A_{\text{s}}| - \sum_{a_i \in A_{\text{med}}} m'(a_i, A_{\text{s}})}{M_W(W, A_{\text{s}}) M_H(H, A_{\text{s}})} \right\rceil\right\}.$$

If $L_{3,\text{BM}}^{2\text{D}}(p, q) = |A_{\text{large}} \cup A_{\text{med}}|$, $L_{3,\text{BM}}^{2\text{D}}$ is dominated by L^{DDFF} because it is dominated by the continuous bound after application of $f_0^{(1/2)W}$ and $f_0^{(1/2)H}$ to the widths and to the heights, respectively.

The remaining case is the following:

$$L^{2D}_{3,BM}(p,q) = |A_{large} \cup A_{med}| + \left\lceil \frac{|A_s| - \sum_{a_i \in A_{med}} m'(a_i, A_s)}{M_W(W, A_s) M_H(H, A_s)} \right\rceil.$$

We now show that f_1^k leads to a bound which is equal to $L^{2D}_{3,BM}$. To simplify the proof, we introduce $W' = M_W(W, A_s)$ and $H' = M_H(H, A_s)$.

$$L^{2D}_{(f_1^p, f_1^q)} = \left\lceil \sum_{a_i \in A} \frac{f_1^q(w_i) \times f_1^p(h_i)}{W'H'} \right\rceil.$$

By replacing $f_1^q(w_i)$ and $f_1^p(h_i)$ by their values, we obtain

$$L^{2D}_{(f_1^p, f_1^q)} = \left\lceil \frac{\sum_{a_i \in A_{large}} W'H' + \sum_{a_i \in A_s} 1}{W'H'} + \frac{\sum_{a_i \in A_{med}} (W' - M_W(W - w_i, A_s))(H' - M_H(H - h_i, A_s))}{W'H'} \right\rceil,$$

$$L^{2D}_{(f_1^p, f_1^q)} = |A_{large}| + \left\lceil \frac{|A_s| + \sum_{a_i \in A_{med}} (W' - M_W(W - w_i, A_s))(H' - M_H(H - h_i, A_s))}{W'H'} \right\rceil,$$

$$L^{2D}_{(f_1^p, f_1^q)} = |A_{large}| + |A_{med}| + \left\lceil \sum_{a_i \in A_{med}} \left(-\frac{H' M_W(W - w_i, A_s)}{W'H'} - \frac{W' M_H(H - h_i, A_s)}{W'H'} + \frac{M_H(H - h_i, A_s) M_W(W - w_i, A_s)}{W'H'} \right) + \frac{|A_s|}{W'H'} \right\rceil = L^{2D}_{3,BM}(p,q),$$

$$L^{2D}_{1,CCM} \geqslant L^{2D}_{(f_1^p, f_1^q)} = L^{2D}_{3,BM}. \qquad \square$$

Theorem 5.3. *$L^{2D}_{1,CCM}$ dominates the bound $L^{2D}_{4,BM}$ proposed in* [2].

Proof. We consider only the case where the bound is different from $|A_{large} \cup A_{med}|$. We show that f_2^k actually leads to a better bound.

$$L^{2D}_{(f_2^p, f_2^q)} = \left\lceil \sum_{a_i \in A} \frac{f_2^q(w_i) \times f_2^p(h_i)}{2\lfloor W/q \rfloor 2\lfloor H/p \rfloor} \right\rceil.$$

In order to simplify the proof, we take into account the fact that f_2^k is an increasing function. Thus, $f_2^k(x) \geqslant 2\lfloor x/k \rfloor$ if $x = C/2$. Replacing the f_2^p and f_2^q by their values, we obtain

$$L^{2D}_{(f_2^p, f_2^q)} \geqslant |A_{\text{large}}| + \left\lceil \sum_{a_i \in A_{\text{med}}} \frac{\left(2\left\lfloor \frac{H}{p} \right\rfloor - 2\left\lfloor \frac{H-h_i}{p} \right\rfloor\right)\left(2\left\lfloor \frac{W}{q} \right\rfloor - 2\left\lfloor \frac{W-w_i}{q} \right\rfloor\right)}{4\left\lfloor \frac{W}{q} \right\rfloor \left\lfloor \frac{H}{p} \right\rfloor} + \sum_{a_i \in A_s^t} \frac{\left(2\left\lfloor \frac{H}{p} \right\rfloor - 2\left\lfloor \frac{H-h_i}{p} \right\rfloor\right) 2\left\lfloor \frac{w_i}{q} \right\rfloor}{4\left\lfloor \frac{W}{q} \right\rfloor \left\lfloor \frac{H}{p} \right\rfloor} + \sum_{a_i \in A_s^w} \frac{2\left\lfloor \frac{h_i}{p} \right\rfloor \left(2\left\lfloor \frac{W}{q} \right\rfloor - 2\left\lfloor \frac{W-w_i}{q} \right\rfloor\right)}{4\left\lfloor \frac{W}{q} \right\rfloor \left\lfloor \frac{H}{p} \right\rfloor} + \sum_{a_i \in A_s^s} \frac{4\left\lfloor \frac{h_i}{p} \right\rfloor \left\lfloor \frac{w_i}{q} \right\rfloor}{4\left\lfloor \frac{W}{q} \right\rfloor \left\lfloor \frac{H}{p} \right\rfloor} \right\rceil,$$

$$L^{2D}_{(f_2^p, f_2^q)} \geqslant |A_{\text{large}}| + |A_{\text{med}}| + \left\lceil \sum_{a_i \in A_{\text{med}}} \frac{-\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W-w_i}{q} \right\rfloor - \left\lfloor \frac{W}{q} \right\rfloor \left\lfloor \frac{H-h_i}{p} \right\rfloor + \left\lfloor \frac{W-w_i}{q} \right\rfloor \left\lfloor \frac{H-h_i}{p} \right\rfloor}{\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W}{q} \right\rfloor} + \sum_{a_i \in A_s^t} \frac{\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{w_i}{q} \right\rfloor - \left\lfloor \frac{H-h_i}{p} \right\rfloor \left\lfloor \frac{w_i}{q} \right\rfloor}{\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W}{q} \right\rfloor} + \sum_{a_i \in A_s^w} \frac{\left\lfloor \frac{W}{q} \right\rfloor \left\lfloor \frac{h_i}{p} \right\rfloor - \left\lfloor \frac{W-w_i}{q} \right\rfloor \left\lfloor \frac{h_i}{p} \right\rfloor}{\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W}{q} \right\rfloor} + \sum_{a_i \in A_s^s} \frac{\left\lfloor \frac{h_i}{p} \right\rfloor \left\lfloor \frac{w_i}{q} \right\rfloor}{\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W}{q} \right\rfloor} \right\rceil,$$

$$L^{2D}_{(f_2^p, f_2^q)} \geqslant |A_{\text{large}}| + |A_{\text{med}}| + \left\lceil \sum_{a_i \in A \setminus A_{\text{large}}} \frac{m''(a_i, p, q)}{\left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W}{q} \right\rfloor} \right\rceil = L^{2D}_{4,\text{BM}}(p, q),$$

$$L^{2D}_{1,\text{CCM}} \geqslant L^{2D}_{(f_2^p, f_2^q)} \geqslant L^{2D}_{4,\text{BM}}. \qquad \square$$

6. Computational experiments

We tested our reduction procedures and our lower bounds on a classical benchmark described by Berkey and Wang [12] and Martello and Vigo [4]. The benchmark is composed of 10 classes of randomly

generated test problems. Each class contains five groups of 10 instances each. These data are available on the following web page: http://www.or.deis.unibo.it/ORinstances. Results for the methods proposed by Boschetti and Mingozzi are known for these benchmarks [13]. All programs are implemented in C on a PC Pentium IV 2.6 GHz running Linux.

6.1. Reduction procedures

We tested our reduction procedures and compared them to the method proposed by Boschetti and Mingozzi [2] (Table 1). The heuristic used for the decision problems inside v_1 and v_2 is based on the well-known *bottom left decreasing* heuristic [19]. We first sort the set of items following a given order. The free areas are considered from left to right, an item is packed in the first free area. We note that a more efficient heuristic could improve the results but we prefer using a fast method.

The comparison criteria for the reduction procedures are:

1. The percentage of items left in the instance obtained after removing the items of size (W, H).
2. The total area of the obtained items divided by the area of a bin.
3. The computing time in seconds.

There are four results for each criterion: one for each reduction procedure, and one for the iterative use of the three reduction procedures, i.e. each function is applied in turn while the instance is modified (column *all*). We also report the initial total area, which helps evaluating the efficiency of the different methods to increase the size of the items.

In most cases, function u is less efficient for removing items. It is sometimes useful to improve on the results of the two other reduction procedures. Function v_1 is the best function on average for reducing the size of the instance. The only class for which there are better results (provided by v_2) is class IX. For classes I, III and VIII, v_1 is the only method which removes items whose size is not equal to the size of the bin initially. Function v_2 is often less efficient than v_1, but it is very interesting to use this function when the items are large, like in Class IX.

For the total area, function u is on average as good as function v_2. Once again, function v_1 is the best method for this criterion. v_1 provides an inferior result when compared to v_2 only for class IX. Most of the time, v_1 returns the best results.

Note that the results for *all* are better than the maximum of the three methods for both criteria. Even when a reduction procedure fails to remove items of the initial instance, it may remove items after application of another procedure. The three procedures should be used together to obtain the best reduction procedure. It appears that the cost of u_2 and v_1 are small, whereas the cost of v_2 is larger. Function v_1 is sufficient for most of the instances.

6.2. Lower bounds

The computation of the lower bounds for 2*BP* can be improved by taking into account the two following relations.

1. If f is a DFF, $f(h_i)/f(H) \leqslant 1/\lfloor H/h_i \rfloor$, as $\lfloor H/h_i \rfloor$ is the number of items of size h_i which can be packed together in a bin of size H.
2. If f is a DDFF, $f(h_i)/f(H) \leqslant 1$, as $f(h_i) \leqslant f(H)$.

Table 1
Reduction procedures

Class	Problem		% Remaining items				Total area/area of a bin					CPU time			
	$H \times W$	n	u	v_1	v_2	all	init	u	v_1	v_2	all	u	v_1	v_2	all
	10 × 10	20	100.00	80.50	98.50	78.00	5.8	5.8	6.0	5.8	6.2	0.0	0.0	0.0	0.0
		40	100.00	63.50	98.50	63.25	11.6	11.6	12.6	11.6	12.7	0.0	0.0	0.0	0.0
		60	100.00	74.83	99.00	67.67	17.9	17.9	18.9	17.9	19.1	0.0	0.0	0.0	0.0
		80	100.00	75.00	99.00	66.38	24.8	24.8	26.0	24.8	26.4	0.0	0.0	0.0	0.0
		100	100.00	84.40	99.00	84.40	30.0	30.0	30.6	30.0	30.6	0.0	0.0	0.0	0.0
	30 × 30	20	100.00	100.00	100.00	100.00	0.6	0.6	0.6	0.6	0.6	0.0	0.0	0.0	0.0
		40	100.00	100.00	100.00	100.00	1.3	1.3	1.3	1.3	1.3	0.0	0.0	0.0	0.0
		60	100.00	100.00	100.00	100.00	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0.0
		80	100.00	100.00	100.00	100.00	2.8	2.8	2.8	2.8	2.8	0.0	0.0	0.0	0.1
		100	100.00	100.00	100.00	100.00	3.3	3.3	3.3	3.3	3.3	0.0	0.0	0.0	0.1
	40 × 40	20	100.00	85.00	100.00	84.50	3.8	3.9	4.0	3.8	4.1	0.0	0.0	0.0	0.0
		40	100.00	85.25	100.00	81.00	7.7	7.7	8.1	7.7	8.2	0.0	0.0	0.0	0.0
		60	100.00	94.67	100.00	92.00	12.0	12.0	12.1	12.0	12.1	0.0	0.0	0.1	0.2
		80	100.00	90.50	100.00	90.25	16.7	16.7	16.9	16.7	17.0	0.0	0.0	0.1	0.2
		100	100.00	98.50	100.00	98.50	20.1	20.1	20.1	20.1	20.1	0.0	0.0	0.2	0.4
	100 × 100	20	100.00	100.00	100.00	100.00	0.6	0.6	0.6	0.6	0.6	0.0	0.0	0.0	0.0
		40	100.00	100.00	100.00	100.00	1.2	1.2	1.2	1.2	1.2	0.0	0.0	0.2	0.4
		60	100.00	100.00	100.00	100.00	1.9	1.9	1.9	1.9	1.9	0.0	0.0	0.5	1.0
		80	100.00	100.00	100.00	100.00	2.7	2.7	2.7	2.7	2.7	0.0	0.0	0.8	1.7
		100	100.00	100.00	100.00	100.00	3.2	3.2	3.2	3.2	3.2	0.0	0.0	1.2	2.5
	100 × 100	20	100.00	83.50	98.50	81.00	4.8	5.0	5.1	4.8	5.4	0.0	0.0	0.0	0.0
		40	100.00	87.50	93.25	75.25	9.7	9.8	10.2	9.8	10.7	0.0	0.0	0.1	0.1
		60	100.00	86.83	100.00	84.00	15.1	15.2	15.6	15.1	15.8	0.0	0.0	0.3	0.4
		80	100.00	79.25	99.88	75.12	21.0	21.0	22.0	21.0	22.3	0.0	0.0	0.5	0.6
		100	100.00	89.80	99.60	89.40	25.3	25.3	25.7	25.3	25.8	0.0	0.0	0.9	1.3
	300 × 300	20	100.00	100.00	100.00	100.00	0.5	0.5	0.5	0.5	0.5	0.0	0.0	0.0	0.1
		40	100.00	100.00	100.00	100.00	1.1	1.1	1.1	1.1	1.1	0.0	0.0	0.4	0.9
		60	100.00	100.00	100.00	100.00	1.7	1.7	1.7	1.7	1.7	0.0	0.0	1.5	3.1
		80	100.00	100.00	100.00	100.00	2.3	2.3	2.3	2.3	2.3	0.0	0.0	3.1	6.3
		100	100.00	100.00	100.00	100.00	2.8	2.8	2.8	2.8	2.8	0.0	0.0	5.6	11.5
	100 × 100	20	100.00	96.00	98.50	96.00	4.2	4.5	4.3	4.3	4.7	0.0	0.0	0.0	0.0
		40	100.00	98.50	99.75	97.75	9.1	9.6	9.2	9.1	9.7	0.0	0.0	0.0	0.1
		60	100.00	97.67	99.67	97.50	13.4	13.6	13.6	13.4	13.9	0.0	0.0	0.1	0.2
		80	100.00	95.75	100.00	95.75	19.3	19.6	19.9	19.3	20.3	0.0	0.0	0.3	0.4
		100	100.00	96.50	100.00	96.50	23.3	23.5	23.9	23.3	24.1	0.0	0.0	0.5	0.8
	100 × 100	20	100.00	97.50	100.00	97.50	4.3	4.7	4.5	4.3	4.9	0.0	0.0	0.0	0.0
		40	100.00	97.25	100.00	97.25	9.2	9.5	9.4	9.2	9.7	0.0	0.0	0.0	0.1
		60	100.00	97.17	100.00	97.17	13.6	13.9	14.0	13.6	14.3	0.0	0.0	0.1	0.2
		80	100.00	97.62	100.00	96.12	19.1	19.2	19.5	19.1	19.9	0.0	0.0	0.3	0.5
		100	100.00	94.50	100.00	94.10	23.6	23.7	24.8	23.6	25.2	0.0	0.0	0.5	0.7

Table 1 (*continued*)

Class	Problem		% Remaining items				Total area/area of a bin					CPU time			
	$H \times W$	n	u	v_1	v_2	all	init	u	v_1	v_2	all	u	v_1	v_2	all
	100×100	20	100.00	57.50	13.00	1.00	8.9	10.7	10.8	13.8	14.3	0.0	0.0	0.0	0.0
		40	100.00	40.50	20.25	5.25	17.6	19.0	23.1	26.0	27.6	0.0	0.0	0.0	0.0
		60	100.00	31.67	6.50	1.83	27.2	28.5	37.7	42.5	43.7	0.0	0.0	0.0	0.0
		80	100.00	28.88	6.88	3.12	36.5	37.2	51.4	55.8	57.5	0.0	0.0	0.0	0.0
		100	100.00	16.30	5.20	2.40	44.5	45.0	64.8	67.4	69.3	0.0	0.0	0.1	0.0
	100×100	20	100.00	76.00	97.00	73.50	3.3	3.3	3.4	3.3	3.5	0.0	0.0	0.0	0.0
		40	100.00	90.00	100.00	90.00	6.3	6.3	6.4	6.3	6.4	0.0	0.0	0.1	0.2
		60	100.00	85.17	100.00	85.00	9.0	9.0	9.2	9.0	9.2	0.0	0.0	0.4	0.6
		80	100.00	93.62	100.00	93.12	11.7	11.7	11.8	11.7	11.8	0.0	0.0	0.8	1.4
		100	100.00	95.60	100.00	95.60	14.7	14.7	14.8	14.7	14.8	0.0	0.0	1.2	2.2

So when a function is applied on one dimension, it can be checked that the residual instance can still improve the value of the current best lower bound. Using this method, many combinations are not considered. When a function f is applied to the width, and if $\lceil \sum_{a_i \in A} (f(w_i)/f(W))(1/\lfloor H/h_i \rfloor) \rceil$ is less than the current best lower bound, only f_1^k is used for the height. If $\lceil \sum_{a_i \in A} f(w_i)/f(W) \rceil$ is less than the current best lower bound, no function is used for the height.

6.2.1. *Computational results for* 1*BP*

As lower bounds for 1*BP* are used in our 2*BP* lower bounds, we first compare our bounds for the one-dimensional case to the bounds proposed by Fekete and Schepers [8]. The benchmarks used are computed using the method described in [17]. We generate problems with a bin of size 100, with n items of size included in a given interval [min, 100]. Several values of n (100, 500, 1000) and min (1, 20, 35) are used. For the method of Fekete and Schepers ($L_{\mathrm{FS}}^{1\mathrm{D}}$) and for our method ($L_{\mathrm{CCM}}^{1\mathrm{D}}$), we report the sum of the lower bounds obtained for the 1000 instances (Sum), the number of times where a bound is better than the other (Best bound), and the computing time needed (CPU Time) in seconds for the 1000 test cases (Table 2).

Most of the time, the two bounds are equal. It appears that for each bound, there are test cases where it is better than the other. In average, our bound is better, but is more time consuming. A way of computing good lower bounds is to use both methods.

6.2.2. *Computational results for* 2*BP*

We tested our lower bounds on the same benchmarks (Tables 3 and 4). Note that the reduction procedures are not used, as they do not improve the results of our lower bounds. Two criteria are reported to evaluate the bounds:

1. The value U/L reported is the ratio between the upper bound proposed by Boschetti and Mingozzi [13] and the tested lower bound. The values for the upper bound were kindly sent by Marco Boschetti.

Table 2
Results for 1*BP*

Interval	n	Sum		Best bound			CPU time	
		L^{1D}_{CCM}	L^{1D}_{FS}	Equal	L^{1D}_{CCM}	L^{1D}_{FS}	L^{1D}_{CCM}	L^{1D}_{FS}
[1, 100]	100	52 603	52 584	973	23	4	1	0
	500	255 589	255 583	946	30	24	10	4
	1000	507 910	507 886	927	47	26	22	9
[20, 100]	100	64 518	64 516	974	14	12	1	0
	500	315 142	315 139	931	37	32	8	4
	1000	626 873	626 836	922	53	25	17	10
[35, 100]	100	78 635	78 635	1000	0	0	1	0
	500	385 782	385 782	1000	0	0	6	5
	1000	768 816	768 816	1000	0	0	13	10

2. *Opt* is the number of instances solved to optimality using the two bounds.
3. CPU Time is the computing time needed for one test case.

In Table 3, we report the results of our lower bounds. 1, *CCM* denotes the bound $L^{2D}_{1,CCM}$ obtained by computing the continuous lower bound after application of our DDFF. *CCM* denotes the bound L^{2D}_{CCM}, and 3, *CCM* denotes the bound $L^{2D}_{3,CCM}$. It appears that $L^{2D}_{2,CCM}$ improves the results for five instances, whereas $L^{2D}_{3,CCM}$ only improves the result for one instance (see Class III). As the latter is far more time consuming than the original method, it should only be used if a large amount of time is allowed (before running an exact method, for example). The additional computational effort for $L^{2D}_{2,CCM}$ is small, so L^{2D}_{CCM} is the more efficient to use within an exact method.

In Table 4, we compare our lower bounds to previous ones. *BM* denotes the bound proposed by Boschetti and Mingozzi [2]. We have tested the bound L^{1D}_{FS} of Fekete and Schepers as a subroutine in the bound $L^{2D}_{2,BM}$, it does not improve on the obtained results. We have also tried to use the functions proposed by Fekete and Schepers for 2*BP*, they do not improve the results either. Columns *BM*, *CCM* and 3, *CCM*, respectively, refer to L^{2D}_{BM}, L^{2D}_{CCM} and $L^{2D}_{3,CCM}$. For all classes, the theoretical dominance is confirmed. L^{2D}_{CCM} is always better than L^{2D}_{BM}. Our bounds lead to improved results for 22 instances of classes I, III, V, VII, VIII, and X. Note that the computing time of L^{2D}_{CCM} is smaller than the computing time of L^{2D}_{BM}. This can be explained by the fact that the fast implementation described above avoids a large number of non useful combinations. These methods can only be applied if the two dimensions are considered separately.

7. Conclusion

We have proposed procedures which reduce the size of the original instance. They can be used together and dramatically simplify the problem for several instances of a well-known benchmark. We have also proposed new lower bounds for 2*BP* and shown that they dominate those obtained previously. The

Table 3
Our lower bounds

Class	Problem		Ratio U/L			Opt			CPU time		
	$H \times W$	n	1,CCM	CCM	3,CCM	1,CCM	CCM	3,CCM	1,CCM	CCM	3,CCM
I	10 × 10	20	1.0125	1.0125	1.0125	9	9	9	0.00	0.00	0.00
		40	1.0153	1.0153	1.0153	8	8	8	0.00	0.00	0.00
		60	1.0104	1.0104	1.0104	8	8	8	0.00	0.00	0.02
		80	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		100	1.0032	1.0032	1.0032	9	9	9	0.00	0.00	0.03
		Avg.	1.0083	1.0083	1.0083	8.8	8.8	8.8	0.000	0.000	0.010
II	30 × 30	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		60	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		80	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		100	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		Avg.	1.0000	1.0000	1.0000	10.0	10.0	10.0	0.000	0.000	0.000
III	40 × 40	20	1.0367	1.0367	1.0200	8	8	9	0.00	0.00	0.00
		40	1.0411	1.0411	1.0411	7	7	7	0.00	0.00	0.05
		60	1.0302	1.0244	1.0244	6	7	7	0.01	0.01	0.17
		80	1.0160	1.0160	1.0160	7	7	7	0.01	0.01	0.37
		100	1.0242	1.0204	1.0204	5	6	6	0.02	0.02	0.93
		Avg.	1.0296	1.0277	1.0244	6.6	7.0	7.2	0.008	0.008	0.304
IV	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		60	1.1000	1.1000	1.1000	8	8	8	0.00	0.00	0.12
		80	1.1000	1.1000	1.1000	7	7	7	0.01	0.01	0.43
		100	1.0333	1.0333	1.0333	9	9	9	0.00	0.00	0.28
		Avg.	1.0467	1.0467	1.0467	8.8	8.8	8.8	0.002	0.002	0.166
V	100 × 100	20	1.0250	1.0000	1.0000	9	10	10	0.00	0.00	0.00
		40	1.0271	1.0271	1.0271	7	7	7	0.00	0.00	0.05
		60	1.0110	1.0110	1.0110	8	8	8	0.01	0.02	0.15
		80	1.0211	1.0211	1.0211	5	5	5	0.05	0.06	1.09
		100	1.0265	1.0265	1.0265	3	3	3	0.13	0.15	3.41
		Avg.	1.0221	1.0171	1.0171	6.4	6.6	6.6	0.038	0.046	0.940
VI	300 × 300	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.3000	1.3000	1.3000	7	7	7	0.00	0.00	0.09
		60	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.01
		80	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.03
		100	1.0667	1.0667	1.0667	8	8	8	0.02	0.03	1.39
		Avg.	1.0733	1.0733	1.0733	9.0	9.0	9.0	0.004	0.006	0.304
VII	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0111	1.0111	1.0111	9	9	9	0.00	0.00	0.01
		60	1.0186	1.0186	1.0186	7	7	7	0.00	0.01	0.14
		80	1.0233	1.0233	1.0233	5	5	5	0.01	0.03	0.52
		100	1.0108	1.0108	1.0108	7	7	7	0.01	0.03	0.64
		Avg.	1.0128	1.0128	1.0128	7.6	7.6	7.6	0.004	0.014	0.262

Table 3 (*continued*)

Class	Problem		Ratio U/L			Opt			CPU time		
	$H \times W$	n	1,CCM	CCM	3,CCM	1,CCM	CCM	3,CCM	1,CCM	CCM	3,CCM
VIII	100 × 100	20	1.0250	1.0000	1.0000	9	10	10	0.00	0.00	0.00
		40	1.0091	1.0091	1.0091	9	9	9	0.01	0.01	0.03
		60	1.0121	1.0121	1.0121	8	8	8	0.03	0.03	0.11
		80	1.0139	1.0139	1.0139	7	7	7	0.06	0.06	0.48
		100	1.0217	1.0217	1.0217	4	4	4	0.14	0.15	1.98
		Avg.	1.0164	1.0114	1.0114	7.4	7.6	7.6	0.048	0.050	0.520
IX	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		60	1.0000	1.0000	1.0000	10	10	10	0.01	0.01	0.00
		80	1.0000	1.0000	1.0000	10	10	10	0.03	0.03	0.02
		100	1.0014	1.0000	1.0000	9	10	10	0.05	0.05	0.04
		Avg.	1.0003	1.0000	1.0000	9.8	10.0	10.0	0.018	0.018	0.012
X	100 × 100	20	1.0700	1.0700	1.0700	8	8	8	0.00	0.00	0.00
		40	1.0343	1.0343	1.0343	8	8	8	0.00	0.00	0.04
		60	1.0450	1.0450	1.0450	6	6	6	0.01	0.01	0.35
		80	1.0563	1.0563	1.0563	3	3	3	0.05	0.05	1.60
		100	1.0584	1.0584	1.0584	1	1	1	0.10	0.11	4.26
		Avg.	1.0528	1.0528	1.0528	5.2	5.2	5.2	0.032	0.034	1.250

Table 4
Comparison with the bound of Boschetti and Mingozzi [2]

Class	Problem		Ratio U/L			Opt			CPU time		
	$H \times W$	n	BM	CCM	3,CCM	BM	CCM	3,CCM	BM	CCM	3,CCM
I	10 × 10	20	1.0268	1.0125	1.0125	8	9	9	0.00	0.00	0.00
		40	1.0153	1.0153	1.0153	8	8	8	0.02	0.00	0.02
		60	1.0227	1.0104	1.0104	6	8	8	0.08	0.00	0.05
		80	1.0042	1.0000	1.0000	9	10	10	0.21	0.00	0.00
		100	1.0065	1.0032	1.0032	8	9	9	0.36	0.00	0.07
		Avg.	1.0151	1.0083	1.0083	7.8	8.8	8.8	0.134	0.000	0.028
II	30 × 30	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0000	1.0000	1.0000	10	10	10	0.01	0.00	0.00
		60	1.0000	1.0000	1.0000	10	10	10	0.06	0.00	0.00
		80	1.0000	1.0000	1.0000	10	10	10	0.15	0.00	0.00
		100	1.0000	1.0000	1.0000	10	10	10	0.30	0.00	0.00
		Avg.	1.0000	1.0000	1.0000	10.0	10.0	10.0	0.104	0.000	0.000
III	40 × 40	20	1.0367	1.0367	1.0200	8	8	9	0.00	0.00	0.00
		40	1.0411	1.0411	1.0411	7	7	7	0.02	0.01	0.12
		60	1.0315	1.0244	1.0244	6	7	7	0.08	0.02	0.41
		80	1.0219	1.0160	1.0160	6	7	7	0.22	0.03	0.89
		100	1.0251	1.0204	1.0204	5	6	6	0.41	0.06	2.17
		Avg.	1.0313	1.0277	1.0244	6.4	7.0	7.2	0.146	0.024	0.718

Table 4 (*continued*)

Class	Problem		Ratio U/L			Opt			CPU time		
	$H \times W$	n	BM	CCM	3,CCM	BM	CCM	3,CCM	BM	CCM	3,CCM
IV	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0000	1.0000	1.0000	10	10	10	0.02	0.00	0.00
		60	1.1000	1.1000	1.1000	8	8	8	0.08	0.01	0.22
		80	1.1000	1.1000	1.1000	7	7	7	0.19	0.02	0.77
		100	1.0333	1.0333	1.0333	9	9	9	0.37	0.01	0.47
		Avg.	1.0467	1.0467	1.0467	8.8	8.8	8.8	0.132	0.008	0.292
V	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0271	1.0271	1.0271	7	7	7	0.04	0.01	0.18
		60	1.0110	1.0110	1.0110	8	8	8	0.14	0.04	0.51
		80	1.0211	1.0211	1.0211	5	5	5	0.38	0.14	3.17
		100	1.0265	1.0265	1.0265	3	3	3	0.67	0.36	9.45
		Avg.	1.0171	1.0171	1.0171	6.6	6.6	6.6	0.246	0.110	2.662
VI	300 × 300	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.3000	1.3000	1.3000	7	7	7	0.02	0.01	0.14
		60	1.0000	1.0000	1.0000	10	10	10	0.08	0.00	0.00
		80	1.0000	1.0000	1.0000	10	10	10	0.21	0.00	0.00
		100	1.0667	1.0667	1.0667	8	8	8	0.41	0.07	2.29
		Avg.	1.0733	1.0733	1.0733	9.0	9.0	9.0	0.144	0.016	0.486
VII	100 × 100	20	1.0367	1.0000	1.0000	8	10	10	0.00	0.00	0.00
		40	1.0202	1.0111	1.0111	8	9	9	0.03	0.00	0.03
		60	1.0257	1.0186	1.0186	6	7	7	0.11	0.02	0.50
		80	1.0367	1.0233	1.0233	2	5	5	0.29	0.07	1.88
		100	1.0188	1.0108	1.0108	5	7	7	0.52	0.07	2.25
		Avg.	1.0276	1.0128	1.0128	5.8	7.6	7.6	0.190	0.032	0.932
VIII	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0091	1.0091	1.0091	9	9	9	0.01	0.01	0.03
		60	1.0184	1.0121	1.0121	7	8	8	0.03	0.02	0.21
		80	1.0139	1.0139	1.0139	7	7	7	0.08	0.05	0.74
		100	1.0217	1.0217	1.0217	4	4	4	0.14	0.12	3.00
		Avg.	1.0126	1.0114	1.0114	7.4	7.6	7.6	0.052	0.040	0.796
IX	100 × 100	20	1.0000	1.0000	1.0000	10	10	10	0.00	0.00	0.00
		40	1.0000	1.0000	1.0000	10	10	10	0.05	0.00	0.00
		60	1.0000	1.0000	1.0000	10	10	10	0.21	0.01	0.01
		80	1.0000	1.0000	1.0000	10	10	10	0.52	0.02	0.02
		100	1.0000	1.0000	1.0000	10	10	10	0.97	0.04	0.04
		Avg.	1.0000	1.0000	1.0000	10.0	10.0	10.0	0.350	0.014	0.014
X	100 × 100	20	1.0700	1.0700	1.0700	8	8	8	0.00	0.00	0.00
		40	1.0343	1.0343	1.0343	8	8	8	0.00	0.00	0.03
		60	1.0672	1.0450	1.0450	4	6	6	0.02	0.01	0.29
		80	1.0640	1.0563	1.0563	2	3	3	0.06	0.04	1.34
		100	1.0584	1.0584	1.0584	1	1	1	0.12	0.09	3.55
		Avg.	1.0588	1.0528	1.0528	4.6	5.2	5.2	0.040	0.028	1.042

theoretical dominance of our lower bounds is confirmed by experiments. They are always better than the previous best bounds and they lead to improved results for several instances. Using a fast implementation, our bounds are faster than the bounds they dominate.

Note that our approach can easily be generalized to the three-dimensional packing problem 3*BP*, following the framework defined by Fekete and Schepers [3]. It can also be applied to variants of 2*BP* such as the problem in which rotation is allowed, or when items have to be packed into levels. We plan to use our reduction procedures and lower bounds within an enumerative method for 2*BP*.

Acknowledgements

The authors would like to thank Emmanuel Néron for his research work and his help for the Discrete DFF.

The authors would also like to thank Marco Boschetti who kindly sent them the results for his lower and upper bounds, and Anis Gharbi, who sent them the benchmarks for 1*BP*.

The authors also thank Khalil Raïs for his computational experiments on the lower bounds.

References

[1] Garey MR, Johnson DS. Computers and intractability, a guide to the theory of NP-completeness. New York: Freeman; 1979.
[2] Boschetti M, Mingozzi A. The two-dimensional finite bin packing problem. Part I: new lower bounds for the oriented case. 4OR 2003;1:27–42.
[3] Fekete S, Schepers J. A general framework for bounds for higher-dimensional orthogonal packing problems. Mathematical Methods of Operations Research 2004;60:311–29.
[4] Martello S, Vigo D. Exact solution of the two-dimensional finite bin packing problem. Management Science 1998;44: 388–99.
[5] Haouari M, Gharbi A. Fast lifting procedures for the bin packing problem. Discrete Optimization 2005, accepted.
[6] Johnson DS. Near optimal bin packing algorithms. Dissertation, Massachussetts Institute of Technology, Cambridge, MA, 1973.
[7] Lueker GS. Bin packing with items uniformly distributed over intervals [a,b]. In: Proceedings of the 24th annual symposium on foundations of computer science (FOCS 83). Silver Spring, MD: IEEE Computer Society; 1983. p. 289–97.
[8] Fekete S, Schepers J. New classes of fast lower bounds for bin packing problems. Mathematical Programming 2001;91: 11–31.
[9] Carlier J, Néron E. Computing redundant resources for the RCPSP. In: Proceedings of the Eighth international workshop on project management and scheduling, Valencia. 2002. p. 92–5.
[10] Caprara A, Locatelli M, Monaci M. Bilinear packing by bilinear programming. IPCO XI, 2005.
[11] Labbé M, Laporte G, Mercure H. Capacitated vehicle routing on trees. Operations Research 1991;39(6):16–22.
[12] Berkey JO, Wang PY. Two-dimensional finite bin-packing algorithms. Journal of Operational Research Society 1987;38: 423–9.
[13] Boschetti M, Mingozzi A. The two-dimensional finite bin packing problem. Part II: new lower and upper bounds. 4OR 2003;1:135–47.
[14] Carlier J, Néron E. A new LP-based lower bound for the cumulative scheduling problem. European Journal of Operational Research 2000;127:363–82.
[15] Bourjolly JM, Rebetez V. An analysis of lower bounds procedures. Computers and Operations Research 2005;32: 395–405.
[16] Martello S, Toth P. Lower bounds and reduction procedure for the bin packing problem. Discrete Applied Mathematics 1990;28:59–70.

[17] Martello S, Toth P. Knapsack problems—algorithms and computer implementation. Chichester: Wiley; 1990.
[18] Boschetti M. New lower bounds for the three-dimensional finite bin packing problem. Discrete Applied Mathematics 2004;140:241–58.
[19] Baker BS, Coffman EG, Rivest RL. Orthogonal packing in two dimensions. SIAM Journal on Computing 1980;9:846–55.