

A survey of dual-feasible and superadditive functions

François Clautiaux^{*}, Cláudio Alves[†], José Valério de Carvalho[†]

^{*} LIFL,UMR CNRS 8022, Université des Sciences et Technologies de Lille, INRIA
Bâtiment INRIA, Parc de la Haute Borne, 59655 Villeneuve d'Ascq, France
`francois.clautiaux@univ-lille1.fr`

[†] Centro de Investigação Algorítmica da Universidade do Minho,
Escola de Engenharia, Universidade do Minho, 4710-057 Braga, Portugal
`{claudio,vc}@dps.uminho.pt`

May 20, 2008

Abstract

Dual-feasible functions are valuable tools that can be used to compute both lower bounds for different combinatorial problems and valid inequalities for integer programs. Several families of functions have been used in the literature. Some of them were defined explicitly, and others not. One of the main objectives of this paper is to survey these functions, and to state results concerning their quality. We clearly identify dominant subsets of functions, *i.e.* those which may lead to better bounds or stronger cuts. We also describe different frameworks that can be used to create dual-feasible functions. With these frameworks, one can get a dominant function based on other ones. Two new families of dual-feasible functions obtained by applying these methods are proposed in this paper.

We also performed a computational comparison on the relative strength of the functions presented in this paper for deriving lower bounds for the bin-packing problem and valid cutting planes for the pattern minimization problem. Extensive experiments on instances generated using methods described in the literature are reported. In many cases, the lower bounds are improved, and the linear relaxations are strengthened.

1 Introduction

The concept of dual-feasible function (DFF) has been used to improve the resolution of several cutting/packing (CP) problems, and more generally any problem involving knapsack inequalities (scheduling problems, vehicle or network routing). It was used for the first time for deriving algorithmic lower bounds for bin-packing problems by Lueker [25]. A subclass of dual-feasible functions restricted to the superadditive and nondecreasing functions have also been used to strengthen cuts [13] for integer programs (see also [1, 27]).

When the literature concerning algorithmic lower bounds for CP problems and the literature concerning valid inequalities for integer programs are considered, it appears that these two literatures are somehow *disconnected*. For example, to our knowledge, a function proposed by Burdett and Johnson [7] in 1977 has never been used for computing lower bounds for bin-packing problems. First the terminology is not the same. Functions designed for lower bounding are often referred to as *dual-feasible*, whereas the functions used to strengthen integer programming models are referred to as *superadditive and nondecreasing*. The relation between the two families of functions is that the latter is a dominant family of DFF. In the context of scheduling problems, Carlier and Néron [12] use the term of *redundant functions*, which are actually discrete DFF. They introduce the notion of *Maximal DFF*, discrete superadditive and nondecreasing functions, with an additional property of *symmetry*.

Superadditive and nondecreasing functions can be used to generate valid inequalities for integer programs (see [27], for example). DFF can also be used to derive valid inequalities; however they have

only been used explicitly with this purpose in [2] and [3]. Many principles that apply to superadditive functions and valid inequalities have a direct counterpart in the theory of dual-feasible functions, but to the best of our knowledge no connections have ever been established.

Most of the methods proposed for computing dual-feasible functions are polynomial and rely on known families of functions (see [25, 16, 17, 9] for example). One can also use enumerative algorithms for computing DFF. Carlier and Néron [12] proposed a branch-and-bound algorithm for computing all discrete dual-feasible functions for a given size of data. For the two-dimensional bin-packing problem, Caprara *et al.* [8] proposed a bilinear program, which computes the *best* pair of DFF to apply to an instance. The method returns excellent results for a reasonable computing time.

Our objective with this paper is to survey the different dual-feasible functions that were used explicitly or not in the literature, to gather new results concerning these DFF and superadditive functions, and to give an insight into the simple frameworks that generally underly complicated formulations. We also generalize several methods of the literature for obtaining more complex DFF from simple ones, and present a new framework that allows to deduce a maximal DFF from a non-maximal DFF. The latter is an important contribution of the paper that allowed us to derive two new families of functions, which improve previous ones.

The problem we used to benchmark the different functions in terms of lower bounding capabilities is the one-dimensional bin-packing problem (1BP). It consists in minimizing the number of identical one-dimensional bins needed to pack a set of small items. It is *NP-hard* [18]. In the typology of [29], it is classified as a *Single Bin-Size Bin-Packing Problem*. This problem has been widely studied, and is an issue in many other problems. Thus computing fast lower bounds for this problem is of great interest. To study the behavior of these DFF when deriving valid inequalities, we used the pattern minimization problem (PMP), which is a lexicographic problem: it consists in finding a solution with a minimum number of different patterns that uses no more than a given number of large objects. This number is obtained by solving first the corresponding cutting stock problem.

We compare existing functions, theoretically, and practically by using them for obtaining lower bounds for the one-dimensional bin-packing problem and valid inequalities for integer programs. Each function is tested alone, and in composition with another function. It transpires from our experiments that gathering the functions described in an integer-programming context, and in an algorithmic context, leads to improved lower bounds for the 1BP and to stronger cuts for the PMP.

In Section 2, we recall the concepts of dual-feasible functions, superadditive nondecreasing functions, and how they can be used to compute lower bounds and strengthen linear formulations. Section 3 is devoted to previous and new techniques designed for obtaining new DFF by combining previously known functions. In Section 4, we analyze functions that have been proposed in the literature, and show that several methods implicitly rely on a dual-feasible function. When dominance relations exist between two families of functions, we emphasize it. Section 5 is dedicated to computational experiments.

2 DFF and superadditive functions

In this section, we introduce some definitions concerning DFF and superadditive functions. We recall several results and, in particular, the fact that *nondecreasing superadditive functions* are a dominant family of *dual-feasible functions*, and that *maximal DFF* [12] are themselves a dominant family of the nondecreasing superadditive functions. We also show that any DFF can be used to generate valid inequalities for integer programs.

2.1 Dual-feasible functions

The concept of dual-feasible function is defined as follows.

Definition 1. *A function $f : [0, 1] \rightarrow [0, 1]$ is dual-feasible if for any finite set S of real numbers, we have*

$$\sum_{x \in S} x \leq 1 \Rightarrow \sum_{x \in S} f(x) \leq 1.$$

An alternative definition can be given when the linear relaxation of the integer program of Gilmore and Gomory [20, 21] is considered. In this program, each item size has a demand b_i , which is equal to the number of times the item must be cut from the initial bins of size C . This program is the following:

$$\min \sum_{p \in P} x_p \quad (1)$$

$$\text{subj. to } \sum_{p \in P} a_{ip} x_p \geq b_i, \quad i = 1, \dots, C \quad (2)$$

$$x_p \geq 0, \quad \forall p \in P \quad (3)$$

$$x_p \text{ integer}, \quad \forall p \in P \quad (4)$$

where P is the set of valid patterns, a_{ip} the number of times item i appears in pattern p , x_p the variable determining how many times the pattern p is used in the solution. The dual of the LP above reads:

$$\max \sum_{i \in I} b_i \pi_i \quad (5)$$

$$\text{subj. to } \sum_{i=1}^C a_{ip} \pi_i \leq 1, \forall p \in P \quad (6)$$

$$\pi_i \geq 0 \quad (7)$$

In this context, a function f is said to be dual-feasible if there exists a valid dual solution π of the linear program of [20, 21] such that $f(i/C) = \pi_i$ for any value i in $[1, C]$. By definition, a bound obtained using a DFF cannot be greater than the linear relaxation of the model of [20, 21], but there is always a function that yields this bound.

Dual-feasible functions are generally defined in $[0, 1]$. However, when data are integer, using discrete values instead may lead to simpler formulations. Carlier and Néron [10, 11, 12] propose a discrete version of DFF. They use the designation of *redundant functions* to denote such functions. These functions are defined from $[0, C]$ to $[0, C']$ (C and C' strictly positive integers) instead of being defined from $[0, 1]$ to $[0, 1]$.

When data of the cutting stock problem are integer, for every discrete DFF, there is an equivalent DFF defined in $[0, 1]$ and vice versa. Let f be a DFF defined from $[0, 1] \cap \mathbb{Q}$ to $[0, 1] \cap \mathbb{Q}$, and $\alpha \in \mathbb{N}$ an integer such that $\alpha f(x) \in \mathbb{N}$ for any $x \in [0, 1] \cap \mathbb{Q}$. A discrete DFF g defined from $[0, C]$ to $[0, \alpha]$ is the following: $g(x) = f(x/C) \times \alpha$.

Throughout this paper, we will use this notation since it generally leads to more-easily understandable formulations.

2.2 Superadditive functions and maximal DFF

Not all dual-feasible functions are interesting. To obtain good lower bounds or strong valid inequalities, one can only use functions that are *superadditive* and nondecreasing. Carlier and Néron introduce the notion of maximal DFF (MDFF) [12], which are not *dominated* by any other valid function. In the sequel, we present these MDFF using a slightly modified notation.

Definition 2. A DFF f is a MDFF if there does not exist any other DFF f' such that $\frac{f(x)}{f(C)} \leq \frac{f'(x)}{f'(C)}$ for all $x \leq C$ and there exists a value y such that $\frac{f(y)}{f(C)} < \frac{f'(y)}{f'(C)}$.

Several properties characterize the MDFF. They have to be nondecreasing, superadditive, and also *symmetric*, i.e. if the sum of two values is C , the sum of their images must be $f(C)$. These properties are stated formally in the following proposition.

Proposition 1. [12] A DFF f is an MDFF if and only if the following conditions hold:

1. f is nondecreasing,
2. f is superadditive, i.e., $f(x) + f(y) \leq f(x + y)$
3. f is symmetric, i.e., $\forall x \in [0, C], f(x) + f(C - x) = f(C)$,

4. $f(0) = 0$.

Most of the functions that are analyzed in this paper are maximal. When they are not, they are generally dominated by another known function. When this is not the case, we provide the new corresponding maximal function.

2.3 DFF and valid inequalities

Apart from computing fast lower bounds, any dual-feasible function can be used to generate valid inequalities for integer programs defined over the sets $S = \{x \in \mathbb{Z}_+^n : \sum_{j=1}^n a_{ij}x_j \leq b_i, i = 1, \dots, m\}$ such that $b_i \geq a_{ij} \geq 0$ and $b_i > 0$ for any i, j . This property is stated formally in the sequel.

Proposition 2. *If $f : [0, 1] \rightarrow [0, 1]$ is a DFF and $S = \{x \in \mathbb{Z}_+^n : \sum_{j=1}^n a_{ij}x_j \leq b_i, i = 1, \dots, m\}$, with $b_i \geq a_{ij} \geq 0$ and $b_i > 0 \forall i, j$, then for any i , $\sum_{j=1}^n f(\frac{a_{ij}}{b_i})x_j \leq 1$ is a valid inequality for S .*

Proof. Given the prerequisites, for any i , we have that $\sum_{j=1}^n \frac{a_{ij}}{b_i}x_j \leq 1$. These inequalities can be rewritten as $\sum_{j=1}^n \sum_{k=1}^{x_j} \frac{a_{ij}}{b_i} \leq 1$. Applying the definition of dual-feasible functions yields the following valid inequality $\sum_{j=1}^n \sum_{k=1}^{x_j} f(\frac{a_{ij}}{b_i}) = \sum_{j=1}^n f(\frac{a_{ij}}{b_i})x_j \leq 1$. \square

Given Proposition 1, this result generalizes (and hence is stronger than) an earlier result presented in Nemhauser and Wolsey [27].

Any valid inequality for S can be obtained either through a superadditive function or is dominated by an inequality that has been computed this way [27]. Cuts that are generated by a superadditive function are commonly referred to as *superadditive inequalities*. Among these cuts, those that are not dominated by any other inequality are said to be maximal. Clearly, the facets of the integer hull of S belong to this set. Maximal valid inequalities are necessarily superadditive, and can be obtained by functions that have the additional property of symmetry. The dominant families of dual-feasible functions are characterized exactly by the same set of properties.

Any maximal inequality for S can be obtained through the Gomory procedure which is based on recursive linear combinations and rounding of other inequalities for S . The problem is that in order to get these maximal cuts, it might be necessary to use a very long recursion. In [27], the basic superadditive function underlying the Gomory procedure was given. Nemhauser and Wolsey made the assumption that other superadditive functions, eventually more complex, might generate these maximal cuts using shorter recursions, hence the relevance of research on dual-feasible functions as tools to compute valid inequalities for integer programs. In other works, alternative characterizations of the integer hull of S in terms of a finite set of superadditive inequalities were proposed [23]. However, even if it is finite, the set of cuts required remains very large.

To the best of our knowledge, the first experiences on using explicitly dual-feasible functions to generate valid inequalities for integer programs was done in [2] and in [3]. The authors applied dual-feasible functions that were used before only for lower bounding in packing problems. The optimization problem addressed in [2] was the PMP. In a previous paper [28], Vanderbeck used a set of superadditive functions (see Section 4.4) to strengthen a column generation model for the PMP. He proved that the resulting valid inequalities were stronger than the rank 1 Chvátal-Gomory cuts, but as shown in [2], the functions that he used are not maximal.

In Section 5, we report on computational experiments on the PMP in order to gain more insight on the ability of the DFF addressed in this paper to generate good valid inequalities for this problem.

3 Frameworks for creating valid DFF

A way of obtaining complex superadditive functions is to *combine* simple superadditive functions. In this section, we first recall some results concerning linear combinations and compositions of functions. Then we describe two frameworks to compute a maximal DFF from another DFF, and we state two other general results. When we analyze specific functions of the literature in Section 4, we will precise on which framework it relies when it is relevant.

3.1 Compositions and linear combinations

A simple way of combining two functions is to compute a linear combination, or to compose two DFF (see [16] for example).

Proposition 3. *A composition, or a positive linear combination of superadditive functions is superadditive.*

More particularly, it is shown in [27] that if f and g are superadditive, then, for $\lambda \geq 0$, λf , $\lfloor f \rfloor$, $f + g$, $\min\{f, g\}$ are superadditive. Note that for $\lambda > 0$, $\max\{0, x - \lambda\}$ is also superadditive, whereas $x + \lambda$ is not.

A composition or a positive linear combination of MDFF is also an MDFF. Function $\min\{f, g\}$ is not maximal, unless $f = g$. As function $f(x) = \lfloor x \rfloor$ is not a MDFF, in general $\lfloor f(x) \rfloor$ is not dominant either.

3.2 Finding a corresponding dominant function by symmetry

In the literature, several superadditive and nondecreasing functions are proposed, which are not maximal. The following result aims at creating a MDFF when one knows a non-maximal superadditive function f . A dominating maximal DFF \hat{f} can be built by keeping the images of the values smaller than $C/2$ and computing the images of the values larger than $C/2$ by symmetry. This is a generalization of what is done implicitly by Carlier *et al.* in [9].

Theorem 1. *Let f be a superadditive and nondecreasing function defined from $[0, C]$ to $[0, f(C)]$, and such that $f(0) = 0$. The following function is a maximal DFF.*

$$\hat{f} : [0, C] \rightarrow [0, 2f(C)]$$

$$x \mapsto \begin{cases} 2f(C) - 2f(C - x), & \text{for } C \geq x > \frac{C}{2}, \\ f(C), & \text{for } x = \frac{C}{2}, \\ 2f(x), & \text{for } x < \frac{C}{2}. \end{cases}$$

Proof. Function \hat{f} is symmetric by construction. By assumption, it is superadditive on $[0, \frac{C}{2}]$. We have to verify that $\hat{f}(x + y) \geq \hat{f}(x) + \hat{f}(y)$ for several cases.

1. If $x, y < \frac{C}{2}$ and $x + y > \frac{C}{2}$. Suppose $\hat{f}(x + y) < \hat{f}(x) + \hat{f}(y)$. It means that $2f(C) - 2f(C - x - y) < 2f(x) + 2f(y)$. It is in contradiction with the fact that f is superadditive.
2. If $x > \frac{C}{2}$ and $y < \frac{C}{2}$. Suppose $\hat{f}(x + y) < \hat{f}(x) + \hat{f}(y)$. It means that $2f(C) - 2f(C - x - y) < 2f(C) - 2f(C - x) + 2f(y)$. We obtain $f(C - x - y) + f(y) > f(C - x)$, which is in contradiction with the fact that f is superadditive.
3. If $x = \frac{C}{2}$ and $y = \frac{C}{2}$, the result is immediate.
4. If $x = \frac{C}{2}$ and $0 < y < \frac{C}{2}$. Suppose $\hat{f}(x + y) < \hat{f}(x) + \hat{f}(y)$. It means that $2f(C) - 2f(\frac{C}{2} - y) < f(C) + 2f(y)$. We obtain $f(C) < 2f(y) + 2f(\frac{C}{2} - y) \leq 2f(\frac{C}{2})$, which is in contradiction with the fact that f is superadditive.

□

Using this result, many well-known simple superadditive functions can lead to MDFF. Note that \hat{f} dominates f , but it is not the only function to dominate f . A slightly weaker result is implicitly used by Carlier and Néron [12] in their enumerative scheme for computing discrete MDFF, but it does not hold when $f(C)$ is odd.

3.3 Improving a function by studying its limiting behavior

Theorem 2 shows another way of obtaining an MDFF from a non-maximal superadditive function f . The case we consider occurs when for some value x where f is not continuous, the value of $f(x)$ can be increased without modifying the other values. This technique leads to an improved function with some singular values x such that $\lim_{\epsilon \rightarrow 0^-} f(x + \epsilon) < f(x) < \lim_{\epsilon \rightarrow 0^+} f(x + \epsilon)$. This technique is implicitly used by [16, 28, 15] for example. Theorem 2 is used in the sequel to show that some functions are maximal.

For the sake of simplicity, for a given function f and a given value x^* for which f is defined, we define \bar{f}^{x^*} the function defined as follows: $\bar{f}^{x^*}(x) = f(x)$ if $x \neq x^*$, and $\bar{f}^{x^*}(x^*) = f(x^*) + \epsilon$ with ϵ a real value as small as needed. Note that when f is a DFF, \bar{f}^{x^*} may or may not be a DFF. In the following theorem, for a given DFF f , I_2 is the set of values x^* such that \bar{f}^{x^*} is also a DFF, *i.e.* the set of values for which the image can be increased. In the sequel we say that a function f is *right-continuous* in x if $\lim_{\epsilon \rightarrow 0^+} f(x + \epsilon) = f(x)$.

Theorem 2. *Let f be a superadditive and nondecreasing function defined from $[0, C]$ to $[0, f(C)]$ such that $f(0) = 0$. We denote by I_2 the subset of values x from $[0, C]$ such that \bar{f}^x is a DFF, and I_1 the set of remaining values. We suppose that I_2 is a discrete set of values $\{x_1, \dots, x_k\}$ and that f is right-continuous on each set $[0, x_1), (x_1, x_2), \dots, (x_k, 1]$ of I_1 .*

For a given function g , the following function

$$h : [0, C] \rightarrow [0, f(C)]$$

$$x \mapsto \begin{cases} f(x) & \text{if } x \in I_1 \\ g(x) & \text{if } x \in I_2 \end{cases}$$

is a superadditive nondecreasing function if the following conditions are true.

1. $f(x) \leq g(x) \leq \lim_{\epsilon \rightarrow 0^+} f(x + \epsilon)$ for any x in I_2
2. $g(x) + g(y) \leq g(x + y)$ if $x, y, x + y \in I_2$
3. $g(x) + f(y) \leq g(x + y)$ if $x \in I_2, x + y \in I_2$ and $y \in I_1$

Proof. By construction, function h is nondecreasing, and such that $h(0) = 0$. We have to show that it is also superadditive, *i.e.*, for any values x and y , $h(x) + h(y) \leq h(x + y)$. The proof consists in three cases, depending on the values x and y .

1. $x \in I_1$ and $y \in I_1$.

If $x + y \in I_1$, the result is true since f is superadditive. If $x + y \in I_2$, by construction $g(x + y) \geq f(x + y) \geq f(x) + f(y)$, so the result is also true in this case.

2. $x \in I_2$ and $y \in I_1$.

If $x + y \in I_2$, the result is true by assumption. If $x + y \in I_1$, we have to show that $g(x) + f(y) \leq f(x + y)$. Since f is superadditive, we have $\lim_{\epsilon \rightarrow 0^+} f(x + \epsilon) + f(y) \leq \lim_{\epsilon \rightarrow 0^+} f(x + y + \epsilon)$. Note that by assumption, $g(x) \leq \lim_{\epsilon \rightarrow 0^+} f(x + \epsilon)$. Function f is right-continuous in $x + y$, so $\lim_{\epsilon \rightarrow 0^+} f(x + y + \epsilon) = f(x + y)$. Consequently, we obtain: $g(x) + f(y) \leq f(x + y)$.

3. $x \in I_2$ and $y \in I_2$.

If $x + y \in I_2$, the result is true by assumption. If $x + y \in I_1$, we have $\lim_{\epsilon \rightarrow 0^+} f(x + \epsilon) + \lim_{\epsilon \rightarrow 0^+} f(y + \epsilon) \leq \lim_{\epsilon \rightarrow 0^+} f(x + y + \epsilon)$. Function f is right-continuous in $x + y$, and by assumption $g(x) \leq \lim_{\epsilon \rightarrow 0^+} f(x + \epsilon)$ and $g(y) \leq \lim_{\epsilon \rightarrow 0^+} f(y + \epsilon)$. Consequently, we obtain the sought result: $g(x) + g(y) \leq \lim_{\epsilon \rightarrow 0^+} f(x + \epsilon) + \lim_{\epsilon \rightarrow 0^+} f(y + \epsilon) \leq \lim_{\epsilon \rightarrow 0^+} f(x + y + \epsilon) = f(x + y)$.

□

3.4 Using two different functions for $\lfloor x \rfloor$ and r_x

In this paragraph, we address the functions that apply on rational values. When x is rational, r_x will denote the fractional part of x ($r_x = x - \lfloor x \rfloor$). Practically speaking, if the data are integer, one can divide all values by a given rational to obtain rational values.

The following result shows that a way of associating two DFF is to apply them separately to the integer part and to the fractional part of the values. This is valid if the conditions of Lemma 1 are verified.

Lemma 1. *Let f and g be two superadditive functions respectively defined on $[0, C]$ and $[0, 1]$. If $f(x+1) - f(x) \geq v^*$ for all $x \in [0, C-1]$, and for all $y, y' \in [0, 1]$ such that $y+y' > 1$, $g(y+y'-1) \geq g(y) + g(y') - v^*$, the function defined as follows*

$$h(x) = f(\lfloor x \rfloor) + g(r_x)$$

is superadditive on $[0, C]$.

Proof. We have to show that $h(x+y) \geq h(x) + h(y)$ for all x, y in $[0, C]$. Two cases have to be considered, depending on the value $r_x + r_y$.

1. If $r_x + r_y < 1$, $\lfloor x+y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$ and $r_{x+y} = r_x + r_y$. The result is immediate.
2. If $r_x + r_y \geq 1$, $\lfloor x+y \rfloor = \lfloor x \rfloor + \lfloor y \rfloor + 1$ and $r_{x+y} = r_x + r_y - 1$. We have $f(\lfloor x \rfloor + \lfloor y \rfloor) \geq f(\lfloor x \rfloor) + f(\lfloor y \rfloor)$. By original assumption, we obtain $f(\lfloor x \rfloor + \lfloor y \rfloor + 1) \geq f(\lfloor x \rfloor) + f(\lfloor y \rfloor) + \min_{i=0, \dots, C-1} \{f(i+1) - f(i)\}$. Now consider function g . We have $g(r_x + r_y - 1) \geq g(r_x) + g(r_y) - g(1)$. The result directly follows from the two relations.

□

The condition of the lemma is restrictive, since only strictly increasing functions f can lead to a superadditive function. However the conditions on function g are not too strong, and many functions can be used. For example, if g is a classical DFF defined from $[0, 1]$ to $[0, 1]$, f has to be strictly increasing and such that $f(x) - f(x-1) > 1$ for all x in $[0, C]$. We will show in the next section that functions proposed in [7] and [24] use this framework.

3.5 Using the ceiling function

The ceiling function is not superadditive. However it can lead to superadditive functions if it is minored by a suitable value. We now generalize several results that use this kind of method.

Lemma 2. *Let f be a superadditive function. If $\beta \geq 1$, $g(x) = \max\{0, \lceil f(x) \rceil - \beta\}$ is superadditive.*

Proof. Consider two values x and y , $x < y$. If $\lceil f(x) \rceil \leq \beta$, since $\lceil \cdot \rceil$ is nondecreasing, we have $g(x+y) \geq g(x) + g(y)$. In the other case, $\lceil f(x+y) \rceil \geq \lceil f(x) \rceil + \lceil f(y) \rceil - 1$. Consequently, $\lceil f(x+y) \rceil - \beta \geq \lceil f(x) \rceil + \lceil f(y) \rceil - 2\beta$. □

This scheme is used implicitly in [28] and [24], and it may lead to interesting results.

4 Comparative analysis of dual-feasible functions

In the sequel, we survey different functions proposed in the literature. We state them explicitly as dual-feasible functions, we give their discrete version, and determine if the function is maximal or not. For the sake of comprehension, we define new notation $f_{X,i}$ for each function, where X is the initial of the authors, and i a number.

4.1 Fekete and Schepers [16]

Fekete and Schepers [16] propose three dual-feasible functions. Two are maximal, but the third is not a MDFF.

The first function f_0^λ is used implicitly by Martello and Toth [26] in their L_2 lower bound for the bin-packing problem. Function f_0^λ , with $\lambda \in [0, \frac{1}{2}]$, consists in removing all values of size less than a given threshold λ , and symmetrically increasing to one the size of the large values.

$$f_0^\lambda : [0, 1] \rightarrow [0, 1]$$

$$x \mapsto \begin{cases} 1, & \text{for } x > 1 - \lambda, \\ x, & \text{for } \lambda \leq x \leq 1 - \lambda, \\ 0, & \text{for } x < \lambda. \end{cases}$$

It has been shown that this function is superadditive and nondecreasing [2, 14], and even maximal [14]. We report neither the equivalent discrete function, nor the proof of maximality, which are straightforward. Only values $k \leq 1/2$ such that $1 - k$ is the size of a large item are interesting. Note that when λ is small enough, f_0^λ is equivalent to the identity function, and so the lower bounds for the bin-packing problem obtained using this function are never smaller than the initial continuous bound.

The second function $f_{FS,1}^\lambda$, $\lambda \in \mathbb{N} \setminus \{0\}$, can be seen as a special rounding procedure. It is an improvement on a function proposed by Lueker [25], using Theorem 2. Values equal to $1/(\lambda + 1)$, $2/(\lambda + 2)$, \dots , $\lambda/(\lambda + 1)$ are not modified.

$$f_{FS,1}^\lambda : [0, 1] \rightarrow [0, 1]$$

$$x \mapsto \begin{cases} x, & \text{for } x(\lambda + 1) \in \mathbb{Z}, \\ \lfloor (\lambda + 1)x \rfloor \frac{1}{\lambda}, & \text{otherwise.} \end{cases}$$

An equivalent discrete DFF $f_{FS,1}^k$, $k \in \mathbb{N}$, states as follows.

$$f_{FS,1}^k : [0, C] \rightarrow [0, Ck]$$

$$x \mapsto \begin{cases} xk, & \text{if } \frac{x(k+1)}{C} \in \mathbb{N}, \\ \lfloor \frac{x(k+1)}{C} \rfloor C, & \text{otherwise.} \end{cases}$$

In [2], a proof is given showing that it is superadditive. The reader can easily check that this function is also symmetric.

Proposition 4. *Function $f_{FS,1}^k$ is a maximal DFF.*

The third function described in [16] is $f_{FS,2}^\lambda$, with $\lambda \in (0, \frac{1}{2}]$.

$$f_{FS,2}^\lambda : [0, 1] \rightarrow [0, 1]$$

$$x \mapsto \begin{cases} 1 - \frac{\lfloor (1-x)\lambda^{-1} \rfloor}{\lfloor \lambda^{-1} \rfloor}, & \text{for } x > \frac{1}{2}, \\ \frac{1}{\lfloor \lambda^{-1} \rfloor}, & \text{for } \lambda \leq x \leq \frac{1}{2}, \\ 0, & \text{for } x < \lambda. \end{cases}$$

This function is not superadditive [2] (for $\lambda < 1/4$, consider $x = \lambda$ and $y = \lambda$). In [16], the authors use f_0^k and $f_{FS,1}^\lambda$ in composition to obtain better results. Other DFF that dominate $f_{FS,2}^\lambda$ have been proposed in the literature [6, 9]. We give next the discrete version of this function, which will make evident that it is dominated by the function $f_{CCM,1}^k$ described below. Let $k \in [1, \frac{C}{2}]$.

$$f_{FS,2}^k : [0, C] \rightarrow [0, \lfloor \frac{C}{k} \rfloor]$$

$$x \mapsto \begin{cases} \lfloor \frac{C}{k} \rfloor - \lfloor \frac{C-x}{k} \rfloor, & \text{for } x > \frac{C}{2}, \\ 1, & \text{for } k \leq x \leq \frac{C}{2}, \\ 0, & \text{for } x < k. \end{cases}$$

4.2 Boschetti and Mingozzi [6]

Boschetti and Mingozzi [6] and Boschetti [4] respectively propose bounds for the two- and three-dimensional bin-packing problems. For the two-dimensional bin-packing problem, they implicitly use function f_0^λ and $f_{BM,1}^k$, an improved discrete version of $f_{FS,2}^k$. We do not report the formulation of this function, as a slightly improved version ([9]) is described next. In [14], it is shown that any iterative composition of $f_0^{k_i}$ and $f_{BM,1}^{j_i}$ is dominated by a function of the form $f_{BM,1}^k \circ f_0^l$.

4.3 Carlier *et al.* [9]

Carlier *et al.* propose a slight improvement on the function of Boschetti [9], by enforcing the image of $\frac{C}{2}$ to be $f_{CCM,1}^k(C)/2$. This function can also be obtained by applying Theorem 1 to function $\lfloor x/k \rfloor$. Note that as for f_0^k , when $k = 1$, this function is equivalent to the identity function. Let $k \in [1, C/2]$.

$$f_{CCM,1}^k : [0, C] \rightarrow \left[0, 2 \left\lfloor \frac{C}{k} \right\rfloor \right]$$

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

Proposition 5. *Function $f_{CCM,1}^k$ dominates $f_{FS,2}^k$.*

Proof. Multiply the expression of $f_{FS,2}^k$ by 2. The values larger than $\frac{C}{2}$ and those less than k have the same image with the two functions, but values between k and $\frac{C}{2}$ have a value equal to $2 \lfloor \frac{x}{k} \rfloor \geq 2$ using $f_{CCM,1}^k$, and 2 using $f_{FS,2}^k \times 2$. Values equal to $\frac{C}{2}$ have an image equal to $\lfloor \frac{C}{k} \rfloor \geq 2$ using $f_{CCM,1}^k$ and 2 using $f_{FS,2}^k \times 2$. Since $f_{CCM,1}^k(x) > 2 \times f_{FS,2}^k(x)$ for any x in $(k, C/2]$, $f_{CCM,1}^k$ dominates $f_{FS,2}^k \times 2$. \square

4.4 Vanderbeck [28]

In [28], Vanderbeck uses a superadditive and nondecreasing function $f_{VB,1}^k$, $k \in \{2, \dots, C\}$, to derive valid inequalities for the pattern minimization problem which are stronger than the rank 1 Chvátal-Gomory cuts [13]. His function states as follows.

$$f_{VB,1}^k : [0, C] \rightarrow [0, k - 1]$$

$$x \mapsto \max\{0, \left\lfloor \frac{kx}{C} \right\rfloor - 1\}.$$

Function $f_{VB,1}^k$ is a DFF, and it is also superadditive (it is direct using Lemma 2). In [2], it is shown that $f_{FS,1}^k$ dominates $f_{VB,1}^k$. This conclusion becomes clear when we rewrite this function similarly to $f_{FS,1}^k$.

$$f_{VB,1}^k : [0, C] \rightarrow [0, k - 1]$$

$$x \mapsto \begin{cases} \frac{kx}{C} - 1, & \text{for } \frac{kx}{C} \in \mathbb{Z}, \\ \lfloor \frac{kx}{C} \rfloor, & \text{otherwise.} \end{cases}$$

It is easy to see that this function is not maximal (for $k = 2$ and $x = C/2$, $f_{VB,1}^k(\frac{C}{2}) + f_{VB,1}^k(\frac{C}{2}) = \frac{k}{2} - 1 + \frac{k}{2} - 1 < k - 1 = f_{VB,1}^k(C)$). Using Theorem 1 and function $f_{VB,1}^k$, we derive the following new DFF $f_{VB,2}^k$ which is maximal. Hence, no dominance relation links the new $f_{VB,2}^k$ with $f_{FS,1}^k$. Let $k \in \{2, \dots, C\}$.

Proposition 6. *The following function is a maximal DFF:*

$$f_{VB,2}^k : [0, C] \rightarrow [0, 2k - 2]$$

$$x \mapsto \begin{cases} 2k - 2, & \text{for } C - \frac{C}{k} \leq x \leq C, \\ 2 \lfloor \frac{kx}{C} \rfloor, & \text{for } \frac{C}{2} < x < C - \frac{C}{k}, \\ k - 1, & \text{for } x = \frac{C}{2}, \\ 2 \times \lceil \frac{kx}{C} \rceil - 2, & \text{for } \frac{C}{k} < x < \frac{C}{2}, \\ 0, & \text{for } 0 \leq x \leq \frac{C}{k}. \end{cases}$$

Proof. Function $f_{VB,1}^k$ is superadditive [28]. By virtue of Theorem 1, function $f_{VB,2}^k$ is maximal and, hence, it dominates $f_{VB,1}^k$. \square

4.5 Burdett and Johnson [7]

In [7], Burdett and Johnson propose a function defined for rational values. When valid inequalities are considered, the data we have to deal with are often rational. If one wants to use this function for bounding (where data are in general integer), one can multiply the initial values by a rational constant. For a given value x , let r_x be the fractional part of x . The function of Burdett and Johnson is given next.

$$f_{BJ,1} : [0, C] \rightarrow [0, \lfloor C \rfloor]$$

$$x \mapsto \lfloor x \rfloor + \max \left\{ 0, \frac{r_x - r_C}{1 - r_C} \right\}.$$

Any value $\alpha \neq 1$ can replace r_C in this function, but it appears [27] that the stronger inequality is obtained when $\alpha = r_C$. If $r_C = 0$, the function is equal to the identity function. In [27], the authors show that this function is superadditive. An alternative immediate proof of superadditivity derives directly from Lemma 1.

It is shown in [27] that this function is superadditive and nondecreasing. The reader can check that it is also symmetric.

Proposition 7. *Function $f_{BJ,1}$ is a maximal DFF.*

We give below the expression of the discrete version of $f_{BJ,1}$. As the initial data are integer, directly applying the function would lead to the identity. Consequently, all values are divided by an integer λ before the function is applied. The multiplier used should not be a divider of C , otherwise the function will not lead to improved results. To simplify the notation, we introduce $\varphi = \lambda - C \pmod{\lambda}$.

$$f_{BJ,1}^\lambda : [0, C] \rightarrow [0, \lfloor \frac{C}{\lambda} \rfloor \varphi]$$

$$x \mapsto \begin{cases} \lfloor \frac{x}{\lambda} \rfloor \varphi, & \text{if } x \pmod{\lambda} \leq C \pmod{\lambda}, \\ \lfloor \frac{x}{\lambda} \rfloor \varphi + x \pmod{\lambda} - C \pmod{\lambda}, & \text{otherwise.} \end{cases}$$

4.6 Letchford and Lodi [24]

Letchford and Lodi [24] propose another way of strengthening Chvátal-Gomory cuts [13] and Gomory fractional cuts [22] in linear programs. In the remainder, we suppose that the fractional part of C is such that $r_C > 0$. In [24], the authors do not precise that their improvement is based on a dual-feasible function. In this paper, we explicitly formulate the dual-feasible function that underlies their method. As $f_{BJ,1}$, it is based on Lemma 1.

Proposition 8. *Let ψ be equal to $\lceil \frac{1}{r_C} \rceil - 1$. The following function is a DFF.*

$$f_{LL,1} : [0, C] \rightarrow [0, (\psi + 1)\lfloor C \rfloor]$$

$$x \mapsto (\psi + 1)\lfloor x \rfloor + \max \left\{ 0, \left\lceil \frac{r_x - r_C}{1 - r_C} \psi \right\rceil \right\}.$$

Clearly this function cannot dominate $f_{BJ,1}$, which is maximal, but it may improve the results for some instances, as we will see in Section 5. As we show in Proposition 9, the superadditivity is due to Lemma 2. In the following proof, we use the fact that $\lceil x + y \rceil = x + \lceil y \rceil$ if x is integer, and the fact that $\lceil x \rceil + \lceil y \rceil \leq \lceil x + y \rceil + 1$ for any x and y .

Proposition 9. *Function $f_{LL,1}$ implicitly used by Letchford and Lodi [24] is superadditive.*

Proof. We use Lemma 1 to show this result. $(\psi + 1) \times x$ plays the role of f and $\max\left\{0, \left\lceil \frac{x-r_C}{1-r_C} \psi \right\rceil\right\}$ is the function g of the lemma.

First we prove that both functions are superadditive. For $f(x) = (\psi + 1) \times x$ the result is immediate. For $g(x) = \max\left\{0, \left\lceil \frac{x-r_C}{1-r_C} \psi \right\rceil\right\}$, we use Lemma 2. By replacing ψ by its value in $\left\lceil \frac{x-r_C}{1-r_C} \psi \right\rceil$, we obtain $\left\lceil \frac{x-r_C}{1-r_C} \left\lceil \frac{1}{r_C} - 1 \right\rceil \right\rceil$. This is equal to $\left\lceil \frac{x}{1-r_C} \left\lceil \frac{1}{r_C} - 1 \right\rceil - \frac{r_C}{1-r_C} \left\lceil \frac{1}{r_C} - 1 \right\rceil \right\rceil$. The constant $\frac{r_C}{1-r_C} \left\lceil \frac{1}{r_C} - 1 \right\rceil$ is greater than $\frac{r_C}{1-r_C} \frac{1-r_C}{r_C}$, which is equal to one. Thus by virtue of Lemma 2, $g(x)$ is superadditive.

Now we verify that the condition of Lemma 1 is verified. For all x , $f(x+1) + f(x) = (\psi + 1)x - (\psi + 1)(x-1) = \psi + 1$. This means that the value v^* is $\psi + 1$. For all $y, y' \in [0, 1]$ such that $y + y' > 1$, $g(y + y' + 1) = \left\lceil \frac{y+y'-1-r_C}{1-r_C} \psi \right\rceil = \left\lceil \frac{y+y'-1+r_C-r_C-r_C}{1-r_C} \psi \right\rceil = \left\lceil \frac{y-r_C+y'-r_C}{1-r_C} \psi \right\rceil - \psi$. This is valid since $\psi \in \mathbb{N}$. We also have $g(y) + g(y') - (\psi + 1) = \left\lceil \frac{y-r_C}{1-r_C} \psi \right\rceil + \left\lceil \frac{y'-r_C}{1-r_C} \psi \right\rceil - \psi - 1 \leq \left\lceil \frac{y-r_C+y'-r_C}{1-r_C} \psi \right\rceil - \psi$. Consequently, using Lemma 1, we proved that $f_{LL,1}$ is superadditive. \square

Function $f_{LL,1}$ is not maximal, since there are cases where it is not symmetric.

Proposition 10. *Function $f_{LL,1}$ used implicitly by Letchford and Lodi [24] is not a maximal DFF.*

Proof. Consider a size of bin equal to $10/3$. In this case, $\psi = \lceil 1/(1/3) \rceil - 1 = 2$. We also have $f(C) = (2+1)\lceil 10/3 \rceil = 9$. Now we compute $f(C/2) + f(C/2) = 2 \times f(5/3) = 2 \times (3\lceil 5/3 \rceil + \max\{0, \lceil \frac{2/3-1/3}{1-1/3} \psi \rceil\})$. This is equal to $2 \times (3 + \lceil 1 \rceil) = 8 < 9$. Consequently $f(C) > f(C/2) + f(C/2)$. \square

This means that one can propose an improved version of this function by applying Theorem 1.

Proposition 11. *The following function is a maximal DFF, and dominates $f_{LL,1}$.*

$$f_{LL,2} : [0, C] \rightarrow [0, 2(\psi + 1)\lceil C \rceil]$$

$$x \mapsto \begin{cases} 2(\psi + 1)\lceil x \rceil + 2 \max\left\{0, \left\lceil \frac{x-r_C}{1-r_C} \psi \right\rceil\right\}, & \text{if } x < C/2, \\ (\psi + 1)\lceil C \rceil, & \text{if } x = C/2, \\ 2f_{LL,1}(C) - 2f_{LL,1}(C - x), & \text{if } x > C/2. \end{cases}$$

Proof. We prove in Proposition 9 that $f_{LL,1}$ is superadditive. By virtue of Theorem 1, $f_{LL,2}$ is maximal. It dominates $f_{LL,1}$ by construction. \square

Note that Dash and Günlük [15] have proved that ψ can be replaced in $f_{LL,1}$ by any integer value k greater than ψ . We will use $f_{LL,1}^k$ for this extension.

For the discrete version of $f_{LL,2}^k$, we consider an integer value of C , and we divide the values of the datum by an integer value λ that is not a divisor of C . Let $k \geq \lceil \frac{\lambda}{C \bmod \lambda} \rceil - 1$ and $\theta = \lceil \frac{x \bmod \lambda - C \bmod \lambda}{\lambda - C \bmod \lambda} k \rceil$. The discrete version of $f_{LL,2}^k$ is given next.

$$f_{LL,2}^{k,\lambda} : [0, C] \rightarrow [0, 2(k+1)\lceil \frac{C}{\lambda} \rceil]$$

$$x \mapsto \begin{cases} 2 \left\lfloor \frac{x}{\lambda} \right\rfloor (k+1), & \text{if } x < C/2 \text{ and } x \bmod \lambda \leq C \bmod \lambda, \\ 2 \left\lfloor \frac{x}{\lambda} \right\rfloor (k+1) + 2\theta, & \text{if } x < C/2 \text{ and } x \bmod \lambda > C \bmod \lambda, \\ (k+1)\lceil \frac{C}{\lambda} \rceil, & \text{if } x = C/2, \\ 2(k+1)\lceil \frac{C}{\lambda} \rceil - 2f_{LL,1}^{k,\lambda}(C - x), & \text{if } x > C/2. \end{cases}$$

4.7 Dash and Günlük [15]

A particular case of the so-called *extended 2-step Mixed-Integer Rounding (MIR)* inequalities of Dash and Günlük [15] leads to a cut that can be obtained by applying a MDFF. This function also dominates $f_{LL,1}^k$, yet it is not equal to $f_{LL,2}^k$. Again, we suppose that the fractional part of C is such that $r_C > 0$.

Proposition 12. *Let k be an integer greater than or equal to $\lceil \frac{1}{r_C} \rceil - 1$. The following function is a DFF.*

$$f_{DG,1}^k : [0, C] \rightarrow [0, (k+1)\lfloor C \rfloor]$$

$$x \mapsto \begin{cases} (k+1)\lfloor x \rfloor + (k+1)\frac{r_x - r_C}{1 - r_C} & \text{if } k\frac{1 - r_x}{1 - r_C} \in \mathbb{N} \text{ and } r_x > r_C, \\ (k+1)\lfloor x \rfloor + \max\left\{0, \left\lceil \frac{r_x - r_C}{1 - r_C} k \right\rceil\right\} & \text{otherwise.} \end{cases}$$

Now we show that this function is maximal, since it is obtained from $f_{LL,1}^k$ using Theorem 2. For this purpose, we have to show the three following results (Propositions 13, 14 and 15), which correspond with the three conditions of Theorem 2.

We use the same notation as in Theorem 2 : I_2 is the set of values such that $k\frac{1 - r_x}{1 - r_C} \in \mathbb{N}$ and $r_x > r_C$, and I_1 is the set of remaining values. We also denote g the function defined for $x \in I_2$ as follows: $g(x) = (k+1)\lfloor x \rfloor + (k+1)\frac{r_x - r_C}{1 - r_C}$. Function $f_{LL,1}^k$ will play the role of function f .

Note that in the remainder, we use several times the following transformation: $\lfloor x - y \rfloor = x - \lceil y \rceil$ and $\lceil x - y \rceil = x - \lfloor y \rfloor$ when x is integer.

Note also that for any $x \in I_2$, since $k\frac{1 - r_x}{1 - r_C} \in \mathbb{N}$ and $r_x > r_C$, we have $k\frac{1 - r_C}{1 - r_C} - k\frac{1 - r_x}{1 - r_C} \in \mathbb{N}$, and then $k\frac{r_x - r_C}{1 - r_C} \in \mathbb{N}$.

Proposition 13. *For any value x in I_2 , $f_{LL,1}^k(x) \leq g(x) \leq \lim_{\epsilon \rightarrow 0^+} f_{LL,1}^k(x + \epsilon)$.*

Proof. First we prove that for $x \in I_2$, $f_{LL,1}^k(x) \leq g(x)$. If $x \in I_2$, since $r_x > r_C$ and $k\frac{r_x - r_C}{1 - r_C} \in \mathbb{N}$, $f_{LL,1}^k(x) = (k+1)\lfloor x \rfloor + \frac{r_x - r_C}{1 - r_C} k$. Since in this case, $\frac{r_x - r_C}{1 - r_C} k \leq (k+1)\frac{r_x - r_C}{1 - r_C}$, we obtain $f_{LL,1}^k(x) \leq g(x)$.

For any value x of I_2 , $\lim_{\epsilon \rightarrow 0^+} f_{LL,1}^k(x + \epsilon) = \lim_{\epsilon \rightarrow 0^+} [(k+1)\lfloor x + \epsilon \rfloor + \left\lceil \frac{r_{x+\epsilon} - r_C}{1 - r_C} k \right\rceil] = \lim_{\epsilon \rightarrow 0^+} [(k+1)\lfloor x + \epsilon \rfloor + \left\lceil \frac{r_x - r_C + \epsilon}{1 - r_C} k \right\rceil]$. Since $k\frac{r_x - r_C}{1 - r_C} \in \mathbb{N}$, we obtain $\lim_{\epsilon \rightarrow 0^+} [(k+1)\lfloor x + \epsilon \rfloor + \frac{r_x - r_C}{1 - r_C} k + \left\lceil \frac{\epsilon}{1 - r_C} \right\rceil] = (k+1)\lfloor x \rfloor + \frac{r_x - r_C}{1 - r_C} k + 1$. Since $0 \leq \frac{r_x - r_C}{1 - r_C} \leq 1$, $\frac{r_x - r_C}{1 - r_C} k + 1 \geq \frac{r_x - r_C}{1 - r_C} (k+1)$ thus $g(x) \leq \lim_{x \rightarrow 0^+} f_{LL,1}^k(x)$ when $x \in I_2$. \square

Proposition 14. *If $x \in I_2, y \in I_2$, and $x + y \in I_2$, $g(x) + g(y) \leq g(x + y)$.*

Proof. We have to prove that $\lfloor x \rfloor + \frac{r_x - r_C}{1 - r_C} + \lfloor y \rfloor + \frac{r_y - r_C}{1 - r_C} \leq \lfloor x + y \rfloor + \frac{r_{x+y} - r_C}{1 - r_C}$.

If $r_x + r_y < 1$, $\lfloor x \rfloor + \lfloor y \rfloor = \lfloor x + y \rfloor$ and $r_x + r_y = r_{x+y}$, so the proof is immediate.

If $r_x + r_y \geq 1$, $g(x) + g(y) = \lfloor x \rfloor + \frac{r_x - r_C}{1 - r_C} + \lfloor y \rfloor + \frac{r_y - r_C}{1 - r_C} = \lfloor x + y \rfloor - 1 + \frac{r_{x+y} + 1 - 2r_C}{1 - r_C} = \lfloor x + y \rfloor + \frac{r_{x+y} - r_C}{1 - r_C}$. \square

Proposition 15. *When $x \in I_2, x + y \in I_2$ and $y \in I_1$, $g(x) + f_{LL,1}^k(y) \leq g(x + y)$.*

Proof. We decompose the proof into two cases.

1. If $r_y \leq r_C$, only the case $r_x + r_y < 1$ is possible, otherwise $r_{x+y} = r_x + r_y - 1$ and thus we would have $r_x + r_y - 1 > r_C$, which is not possible, since $r_x < 1$ and $r_y \leq r_C$.

In this case, $f_{LL,1}^k(y) = (k+1)\lfloor y \rfloor$. Consequently, we have to show that $(k+1)\lfloor x \rfloor + (k+1)\frac{r_x - r_C}{1 - r_C} + (k+1)\lfloor y \rfloor \leq (k+1)\lfloor x + y \rfloor + (k+1)\frac{r_{x+y} - r_C}{1 - r_C}$. Since in this case $\lfloor x \rfloor + \lfloor y \rfloor = \lfloor x + y \rfloor$, it can be rewritten as $\frac{r_x - r_C}{1 - r_C} \leq \frac{r_{x+y} - r_C}{1 - r_C}$, which is true, since $r_{x+y} > r_x$ by assumption.

2. If $r_y > r_C$, once again only the case $r_x + r_y < 1$ is possible. Suppose $r_x + r_y \geq 1$. By initial assumption, $y \in I_1$: since $r_y > r_C$, it would mean that $k\frac{1 - r_y}{1 - r_C} \notin \mathbb{N}$ and $k\frac{1 - r_{x+y}}{1 - r_C} = k\frac{1 - (r_x + r_y - 1)}{1 - r_C} = k\frac{1 - r_x}{1 - r_C} + k\frac{1 - r_y}{1 - r_C}$. This contradicts the initial assumption that $k\frac{1 - r_{x+y}}{1 - r_C} \in \mathbb{N}$ and $k\frac{1 - r_x}{1 - r_C} \in \mathbb{N}$.

In this case, the inequality $(k+1)\lfloor x \rfloor + (k+1)\frac{r_x - r_C}{1 - r_C} + (k+1)\lfloor y \rfloor + \max\{0, \lceil \frac{r_y - r_C}{1 - r_C} k \rceil\} \leq (k+1)\lfloor x + y \rfloor + (k+1)\frac{r_{x+y} - r_C}{1 - r_C}$ leads to $(k+1)\frac{r_x - r_C}{1 - r_C} + \lceil \frac{r_y - r_C}{1 - r_C} k \rceil \leq (k+1)\frac{r_{x+y} - r_C}{1 - r_C}$. This is equivalent to

$\left\lceil k \frac{r_y - r_C}{1 - r_C} \right\rceil \leq (k+1) \frac{r_y}{1 - r_C}$. We have $(k+1) \frac{r_y}{1 - r_C} = k \frac{r_y - r_C}{1 - r_C} + k \frac{r_C}{1 - r_C} + \frac{r_y}{1 - r_C}$. Since $r_y > r_C$, this is greater than $k \frac{r_y - r_C}{1 - r_C} + (k+1) \frac{r_C}{1 - r_C}$. By definition, k is greater than $\frac{1}{r_C} - 1$, thus the previous expression is greater than $k \frac{r_y - r_C}{1 - r_C} + \frac{1}{r_C} \frac{r_C}{1 - r_C} = k \frac{r_y - r_C}{1 - r_C} + \frac{1}{1 - r_C} > k \frac{r_y - r_C}{1 - r_C} + 1 \geq \left\lceil k \frac{r_y - r_C}{1 - r_C} \right\rceil$. \square

Proposition 16. $f_{DG,1}^k$ is a superadditive and nondecreasing function.

Proof. We prove this result using Theorem 2. For $f_{DG,1}^k$, function f of Theorem 2 is function $f_{LL,1}^k$, and function g is $(k+1)[x] + (k+1) \frac{r_x - r_C}{1 - r_C}$. I_1 and I_2 are defined as above.

We have proved that $f_{LL,1}^k$ is superadditive and nondecreasing. Now we have to prove that $f_{LL,1}^k$ is piecewise right-continuous on I_1 . If $x \in I_1$ and $r_x \leq r_C$, $\lim_{\epsilon \rightarrow 0^+} f_{LL,1}^k(x + \epsilon) = \lim_{\epsilon \rightarrow 0^+} (k+1)[x + \epsilon] = (k+1)[x] = f_{LL,1}^k(x)$. If $x \in I_1$ and $r_x > r_C$, $\lim_{\epsilon \rightarrow 0^+} f_{LL,1}^k(x + \epsilon) = \lim_{\epsilon \rightarrow 0^+} [(k+1)[x + \epsilon] + \lceil \frac{r_x + \epsilon - r_C}{1 - r_C} k \rceil]$. By definition of I_1 , and since $r_x > r_C$, we know that $\frac{r_x - r_C}{1 - r_C} k \notin \mathbb{N}$. Consequently, this is equal to $(k+1)[x] + \lceil \frac{r_x - r_C}{1 - r_C} k \rceil = f_{LL,1}^k(x)$.

The three conditions of Theorem 2 are shown in the Propositions 13, 14 and 15. \square

Proposition 17. Function $f_{DG,1}^k$ is symmetric and hence is a MDFF.

Proof. First we show that if $x \in I_2$, then $C - x \in I_2$. If $r_x > r_C$, $r_x + r_{C-x} = r_C + 1$ and thus $r_{C-x} > r_C$. We also have $k \frac{r_C - x - r_C}{1 - r_C} = k \frac{1 + r_C - r_x - r_C}{1 - r_C} = k + k \frac{r_C - r_x}{1 - r_C}$. Since k is integer, $k \frac{r_C - r_x}{1 - r_C} \in \mathbb{N}$ implies $k \frac{r_C - x - r_C}{1 - r_C} \in \mathbb{N}$. Consequently, the proof consists of two cases, depending on the value x only.

1. $x \in I_1$ (and $C - x \in I_1$).

If $r_x + r_{C-x} < 1$, $r_x < r_C$ and $r_{C-x} \leq r_C$. In this case, $f_{LL,1}^k(x) + f_{LL,1}^k(C - x) = (k+1)[x] + (k+1)[C - x] = (k+1)[C]$.

If $r_x + r_{C-x} \geq 1$, $r_x + r_{C-x} = r_C + 1$, $r_x > r_C$ and $r_{C-x} > r_C$. In this case, $f_{LL,1}^k(x) + f_{LL,1}^k(C - x) = (k+1)[x] + \lceil \frac{r_x - r_C}{1 - r_C} k \rceil + (k+1)[C - x] + \lceil \frac{r_C - x - r_C}{1 - r_C} k \rceil$. This is equal to $(k+1)[C] - (k+1) + \lceil \frac{r_x - r_C}{1 - r_C} k \rceil + \lceil \frac{r_C - r_x + 1 - r_C}{1 - r_C} k \rceil = (k+1)[C] - (k+1) + k + \lceil \frac{r_x - r_C}{1 - r_C} k \rceil + \lceil \frac{r_C - r_x}{1 - r_C} k \rceil$. Now we use the fact that for any value $a \notin \mathbb{Z}$, $\lceil a \rceil + \lceil -a \rceil = 1$. By assumption $\frac{r_C - r_x}{1 - r_C} k$ is not integer, thus the expression is equal to $(k+1)[C] - (k+1) + k + 1 = f_{LL,1}^k(C)$.

2. $x \in I_2$ (and $C - x \in I_2$).

In this case, $r_x > r_C$, $r_{C-x} > r_C$, and $r_x + r_{C-x} = r_C + 1$. We have $g(x) + g(C - x) = (k+1)[x] + (k+1) \frac{r_x - r_C}{1 - r_C} + (k+1)[C - x] + (k+1) \frac{r_C - x - r_C}{1 - r_C}$. This is equal to $(k+1)[C] - (k+1) + (k+1) \frac{r_x - r_C}{1 - r_C} + (k+1) \frac{r_C - r_x + 1 - r_C}{1 - r_C} = (k+1)[C] - (k+1) + (k+1) \frac{r_x - r_C}{1 - r_C} + (k+1) \frac{r_C - r_x}{1 - r_C} + (k+1) = (k+1)[C] = f_{LL,1}^k(C)$.

Since Proposition 16 states that $f_{DG,1}^k$ is nondecreasing and superadditive, it is a MDFF. \square

Function $f_{LL,2}^k$ and $f_{DG,1}^k$ are not the same. Consider any value x of I_2 less than $C/2$. In this case $f_{LL,2}^k(x) = f_{LL,1}^k(x)$, and $f_{DG,1}^k(x) > f_{LL,1}^k(x)$. Function $f_{LL,2}^k$ is obtained using Theorem 1, which only modifies images of values greater than or equal to $C/2$. Function $f_{DG,1}^k$ is obtained by applying Theorem 2.

Finally, we report the discrete formulation of $f_{DG,1}^k$. Let k and θ defined as in the definition of function $f_{LL,2}^{k,\lambda}$. We keep the notation $\varphi = \lambda - C \pmod{\lambda}$ introduced above. Value x is in I_2 if $(k(\lambda - x \pmod{\lambda})) \pmod{\varphi} = 0$ and $x \pmod{\lambda} > C \pmod{\lambda}$.

$$f_{DG,1}^{k,\lambda} : [0, C] \rightarrow [0, \varphi(k+1) \lfloor \frac{C}{\lambda} \rfloor]$$

$$x \mapsto \begin{cases} \lfloor \frac{x}{\lambda} \rfloor (k+1)\varphi, & \text{if } x \pmod{\lambda} \leq C \pmod{\lambda}, \\ \lfloor \frac{x}{\lambda} \rfloor (k+1)\varphi + \theta\varphi, & \text{if } x \pmod{\lambda} > C \pmod{\lambda}, \\ \lfloor \frac{x}{\lambda} \rfloor (k+1)\varphi + (k+1)(x \pmod{\lambda} - C \pmod{\lambda}), & \text{if } x \in I_2. \end{cases}$$

4.8 Summary

In this paragraph, we sum up the different results stated in the current section. We recall several kind of results: dominance, maximality, and the different techniques underlying each family of functions.

In Table 1, we report a classification of the different functions. For each function, we recall the paper in which context it was proposed (*t.p.* means that it is proposed for the first time in the present paper, and ‘-’ means that it is a trivial DFF). We also report the type of application for which it has originally been designed (*lb* for lower bounding, and *cuts* for improved valid inequalities). Then we give informations for each function: if it is a MDFF, if it explicitly uses Lemmas 1 and 2, or Theorem 1.

Functions $\lfloor \frac{x}{k} \rfloor$ ($k \neq 1$), $f_{FS,2}^k$, $f_{LL,1}^k$ and $f_{VB,1}^k$ are not maximal, whereas $f_{BM,1}^k$ is *almost* maximal. Only the image of $\frac{C}{2}$ makes the latter non-maximal. All the other functions are maximal. Among all the functions considered, only $f_{FS,2}^k$ is not superadditive.

Functions $f_{BJ,1}^k$, $f_{LL,1}^k$ and its improved versions $f_{DG,1}^k$, $f_{LL,2}^k$ are the only ones to use Lemma 1. Note that these functions were originally proposed to derive cuts, which can explain the fact that the fractional part is treated apart.

Lemma 2 is used to modify the fractional part in $f_{LL,1}^k$, $f_{LL,2}^k$ and $f_{DG,1}^k$. It underlies function $f_{VB,1}^k$, and the function that dominates it, $f_{VB,2}^k$. Function $f_{FS,1}^k$ does not use explicitly Lemma 2, but its structure is close to it. As soon as the value $f(x)$ returned by a given function f is dependent on the integrity of x , we consider that function f uses Lemma 2.

Theorem 1 is used by $f_{CCM,1}^k$, $f_{LL,2}^k$ and $f_{VB,2}^k$, and *almost* by $f_{BM,1}^k$.

Theorem 2 is used by $f_{FS,1}^k$ and $f_{DG,1}^k$ to obtain a maximal function, and by $f_{VB,1}^k$ to obtain a non-maximal function.

Function	Paper	Application	MDFF	lem. 1	lem. 2	thm. 1	thm. 2
identity	-	-	yes	no	no	no	no
f_0^λ	[16]	lb	yes	no	no	no	no
$\lfloor \frac{x}{k} \rfloor$	-	-	no	no	no	no	no
$f_{FS,2}^k$	[16]	lb	no	no	no	no	no
$f_{BM,1}^k$	[6]	lb	almost	no	no	almost	no
$f_{CCM,1}^k$	[9]	lb	yes	no	no	yes	no
$f_{VB,1}^k$	[28]	cuts	no	no	yes	no	yes
$f_{FS,1}^k$	[16]	lb	yes	no	almost	no	yes
$f_{BJ,1}^k$	[7]	cuts	yes	yes	no	no	no
$f_{LL,1}^k$	[24]	cuts	no	yes	yes	no	no
$f_{DG,1}^k$	[15]	cuts	yes	yes	yes	no	yes
$f_{LL,2}^k$	t.p.	-	yes	yes	yes	yes	no
$f_{VB,2}^k$	t.p.	-	yes	no	yes	yes	no

Table 1: Summary of the properties of the functions analyzed in this paper

In Section 5, we evaluate the gap between dominated functions and the functions that dominate them, and we compare the effectiveness of the maximal functions.

5 Computational experiments

5.1 Bin-packing instances

In this section, we compare the different DFF analyzed in this paper against several types of instances for the one-dimensional bin-packing problem generated in a classical way. A bin packing instance is a pair (C, I) , where $C \in \mathbb{N}^+$ is the size of the bin, and I a set of n items i , each of size $c_i \in \mathbb{N}$.

We use the functions without any kind of preprocessing, or additional optimization methods. The function is applied to the instance, and then the continuous lower bound L_0 is computed for the instance obtained. $L_0 = \lceil \sum_{i \in I} c_i / C \rceil$.

The instances used are computed using the method described in [26]. We generated problems with a bin of size 100, and n items with sizes in the interval $[\min, 100]$. Several values of n (100, 500, 1000)

and $\min(1, 20, 35)$ are used. For each pair of values, we generated 1000 instances. For instances *I*, *II* and *III*, the values of \min are respectively 1, 20 and 35. Instances 1, 2, 3 are respectively related to $n = 1000$, $n = 100$ and $n = 500$.

For each family of functions and for each set of instances (*Set*), we report the sum of the lower bounds obtained for the 1000 instances (*Sum*), the number of times each bound is equal to the best bound (*Best*), and the number of times each bound is the only one to lead to the best bound (*Only*).

In the last subsection, we compare the DFF in terms of the strength of the valid inequalities that they generate for the PMP.

In order to evaluate the quality of the bounds, we first report the results of the continuous bound (*id*), and the bound obtained using f_0^λ . These results are listed in Table 2. Column *Only* is reported for f_0^λ only, as it is always at least as good as the identity.

Set	Sum		Best		Only
	id	f_0^λ	id	f_0^λ	f_0^λ
I-1	500893	507410	89	1000	911
I-2	50508	52500	180	1000	820
I-3	250644	255245	88	1000	912
II-1	595629	626215	0	1000	1000
II-2	59961	64378	0	1000	1000
II-3	298112	314692	0	1000	1000
III-1	670634	767930	0	1000	1000
III-2	67522	78456	0	1000	1000
III-3	335502	385163	0	1000	1000

Table 2: Comparing the identity function with f_0^λ

The bound obtained with f_0^λ for all valid values of λ is equal to the L_2 bound of Martello and Toth [26]. In the sequel, we will emphasize (in bold) the cases in which a function returns bounds that are better than L_2 in average.

5.1.1 Influence of maximality on the lower bounds

In Tables 3, 4 and 5, we compare the results obtained by some functions before and after applying techniques to obtain maximal functions. The purpose is to give an idea of the strength of each improvement for these functions. In the three tables, only maximal functions have a column *Only*.

In Table 3, we compare the three versions of the rounding functions proposed by Fekete and Schepers [16] ($f_{FS,2}^k$), Boschetti and Mingozzi [5] ($f_{BM,1}^k$) and Carlier *et al.* [9] ($f_{CCM,1}^k$). In Table 4, we report results of Fekete and Schepers ($f_{FS,1}^k$) [16], Vanderbeck ($f_{VB,1}^k$) [28], and of the new function $f_{VB,2}^k$ proposed in this paper. Finally, we compare $f_{LL,1}^k$, $f_{LL,2}^k$ and $f_{DG,1}^k$ in Table 5.

Set	Sum			Best			Only
	$f_{FS,2}^k$	$f_{BM,1}^k$	$f_{CCM,1}^k$	$f_{FS,2}^k$	$f_{BM,1}^k$	$f_{CCM,1}^k$	$f_{CCM,1}^k$
I-1	507156	507688	507811	760	907	1000	93
I-2	52314	52507	52534	819	973	1000	27
I-3	255023	255427	255506	733	924	1000	76
II-1	626165	626531	626712	757	884	1000	116
II-2	64303	64394	64434	878	960	1000	40
II-3	314649	314908	315027	761	895	1000	105
III-1	768816	768816	768816	1000	1000	1000	0
III-2	78624	78624	78624	1000	1000	1000	0
III-3	385782	385782	385782	1000	1000	1000	0

Table 3: Comparing functions $f_{FS,2}^k$, $f_{BM,1}^k$ and $f_{CCM,1}^k$

As expected, maximal functions lead to improved results compared to non-maximal functions. What is more surprising is the fact that function $f_{CCM,1}^k$ is strictly better than $f_{BM,1}^k$ for many test cases,

Set	Sum			Best			Only	
	$f_{VB,1}^k$	$f_{FS,1}^k$	$f_{VB,2}^k$	$f_{VB,1}^k$	$f_{FS,1}^k$	$f_{VB,2}^k$	$f_{FS,1}^k$	$f_{VB,2}^k$
I-1	504425	506153	504704	466	966	546	454	31
I-2	51797	52095	51851	713	988	766	234	10
I-3	253341	254397	253555	488	968	576	424	32
II-1	621066	624635	622564	255	941	402	598	59
II-2	63341	63859	63512	511	980	671	329	20
II-3	311658	313604	312528	287	936	481	519	64
III-1	758363	763392	758363	236	1000	236	764	0
III-2	76926	77472	76926	556	1000	556	444	0
III-3	380027	382437	380027	339	1000	339	661	0

Table 4: Comparing functions $f_{VB,1}^k$, $f_{FS,1}^k$ and $f_{VB,2}^k$

Set	Sum			Best			Only	
	$f_{LL,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$	$f_{LL,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$
I-1	504653	504675	504676	971	992	993	7	8
I-2	51619	51628	51625	991	1000	997	3	0
I-3	253292	253312	253314	970	990	992	8	10
II-1	622876	622876	622876	1000	1000	1000	0	0
II-2	63404	63404	63404	1000	1000	1000	0	0
II-3	312543	312543	312543	1000	1000	1000	0	0
III-1	750279	750279	750279	1000	1000	1000	0	0
III-2	75426	75426	75426	1000	1000	1000	0	0
III-3	375266	375266	375266	1000	1000	1000	0	0

Table 5: Comparing functions $f_{LL,1}^k$, $f_{LL,2}^k$ and $f_{DG,1}^k$

although it only modifies the image of $\frac{C}{2}$. This can be explained by the fact that the instances are generated randomly, and thus items of size $\frac{C}{2}$ may appear several times in an instance.

Concerning the improvements on $f_{VB,1}^k$, an interesting fact is that the results are improved by a wide range when $f_{FS,1}^k$ is used, although $f_{VB,2}^k$ only leads to some improvements. Nevertheless $f_{VB,2}^k$ can be better than $f_{FS,1}^k$ in several cases.

Compared to $f_{LL,1}^k$, the number of improved values when $f_{LL,2}^k$ or $f_{DG,1}^k$ are used is small. This means that $f_{LL,1}^k$ is *almost* maximal (*i.e.* there are only a few number of pairs $(x, C - x)$ that are not symmetric).

5.1.2 Comparison between the different classes of DFF

From the previous tables, we can see that only the bounds based on the rounding function ($f_{BM,1}^k$ and $f_{CCM,1}^k$) are better than f_0^λ on average. This means that if one wants to use a unique function, $f_{CCM,1}^k$ would be this one. But if one is looking for the best results, he will have to use all the maximal DFF described in this paper.

We sum up the results obtained by all the functions described in this paper in Table 6. We report respectively the number of times the functions lead to the best bound, and the number of times a function is the only one to return the best bound. The functions chosen are those that are not dominated by any other functions. The columns *Sum* are not reported, as they are already reported in previous tables.

Functions f_0^λ and $f_{CCM,1}^k$ yield apparently the best results in average for these instances. Note that each surveyed maximal function leads at least once to a bound that is not attained by the other functions.

5.1.3 Composing functions

We computed the bounds yielded by the functions obtained by composing f_0^λ or $f_{CCM,1}^k$ with the functions tested above. All relevant values of λ are used, so the complexity of the bounds is larger, and consequently

Set	Best							Only						
	f_0^λ	$f_{FS,1}^k$	$f_{VB,2}^k$	$f_{CCM,1}^k$	$f_{BJ,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$	f_0^λ	$f_{FS,1}^k$	$f_{VB,2}^k$	$f_{CCM,1}^k$	$f_{BJ,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$
I-1	548	345	162	826	173	267	267	12	12	2	228	5	2	0
I-2	865	558	399	915	416	453	453	15	3	0	57	6	0	0
I-3	654	378	196	846	229	301	306	25	8	2	162	3	1	1
II-1	513	283	83	822	103	227	227	22	11	4	275	7	0	0
II-2	806	469	267	870	346	415	415	16	7	3	66	5	0	0
II-3	566	308	110	815	140	271	271	31	20	1	204	6	0	0
III-1	515	198	0	1000	0	0	0	0	0	0	485	0	0	0
III-2	826	342	130	1000	58	60	60	0	0	0	139	0	0	0
III-3	577	200	0	1000	0	0	0	0	0	0	423	0	0	0

Table 6: Summary

the computing time is also significantly larger. As f_0^1 is equal to the identity function, the results are always larger than or equal to the previous results. In the subsequent tables columns *Only* have a different meaning. They now report the number of times each function is the only best, and better than the corresponding non-composed function. The results are reported in Tables 7 and 8, we compare these compositions with the compositions obtained using f_0^λ . We did not try to compose $f_{CCM,1}^k$ with itself, as it cannot lead to improved results (see [14]). We do not report results of $f_{DG,1}^k$, since it does not improve the results obtained using $f_{LL,2}^k$.

Set	Sum									
	comp. with f_0^λ					comp. with $f_{CCM,1}^k$				
	$f_{CCM,1}^k$	$f_{FS,1}^k$	$f_{BJ,1}^k$	$f_{VB,2}^k$	$f_{LL,2}^k$	$f_{FS,1}^k$	$f_{BJ,1}^k$	$f_{VB,2}^k$	$f_{LL,2}^k$	
I-1	507956	508000	507923	508015	508091	507954	507903	507989	507844	
I-2	52617	52622	52599	52642	52645	52601	52572	52612	52540	
I-3	255625	255652	255601	255666	255701	255634	255582	255635	255531	
II-1	626958	626976	626869	626984	627076	626900	626813	626924	626725	
II-2	64552	64578	64545	64589	64609	64524	64475	64540	64478	
II-3	315199	315243	315155	315242	315330	315186	315107	315190	315076	
III-1	768816	768816	768816	768816	768816	768816	768816	768816	768444	
III-2	78635	78635	78635	78635	78635	78635	78635	78633	78397	
III-3	385782	385782	385782	385782	385782	385782	385782	385782	385543	

Table 7: Compositions: sum of the bound for each function

Set	Best/Only									
	comp. with f_0^k					comp. with $f_{CCM,1}^k$				
	$f_{CCM,1}^k$	$f_{FS,1}^k$	$f_{BJ,1}^k$	$f_{VB,2}^k$	$f_{LL,2}^k$	$f_{FS,1}^k$	$f_{BJ,1}^k$	$f_{VB,2}^k$	$f_{LL,2}^k$	
I-1	825	3	849	2	796	1	855	8	927	22
I-2	938	2	943	0	920	0	963	4	966	5
I-3	858	4	883	0	837	2	894	8	928	13
II-1	826	5	831	0	763	2	824	5	917	33
II-2	907	2	933	1	900	1	944	7	964	6
II-3	809	3	843	0	772	1	837	7	924	27
III-1	1000	0	1000	0	1000	0	1000	0	1000	0
III-2	1000	0	1000	0	1000	0	1000	0	1000	0
III-3	1000	0	1000	0	1000	0	1000	0	1000	0

Table 8: Compositions: number of times a function leads to the best bound

Composing these functions with f_0^λ leads to improved results for several cases. Each function returns the best lower bound for more than 90% of the instances. An interesting fact is that when composed

with f_0^λ , $f_{CCM,1}^k$ is no more the most interesting function. Now the best bounds seem to be obtained when composing $f_{LL,2}^k$ with f_0^λ .

The computing time entailed by the composition is far larger than the time needed to apply one function only. It takes 4.28 seconds to a computer with a dual core 1.2GHz processor to compute a lower bound for each of the 1000 instance of a benchmark for $f_{CCM,1}$ and 156.28 seconds for the composition of f_0 and $f_{CCM,1}$ on the same computer.

Composing a function with $f_{CCM,1}^k$ is less interesting than with f_0^λ . This can be due to the fact that the instance obtained after applying $f_{CCM,1}^k$ has a smaller number of different values, and cannot be improved by a wide range, whereas an instance modified by f_0^λ may have many different values. However, some bounds are improved using these compositions.

5.1.4 Summary

It transpires that if one has to choose one function to compute lower bounds, $f_{CCM,1}^k$ is the best choice. All maximal functions lead to a bound that is not attained by the mean of any of the other functions. Consequently if one wants to obtain as strong bounds as possible, one should use all families of maximal functions described in this paper.

Non-maximal functions may lead to interesting results, but each time we used the corresponding maximal function instead, the results were strictly improved for several instances. As the computational effort is the same, there is no reason why one would use a non-maximal function.

Composing the functions helps improving the results, at the expense of a far larger computational effort. Composing a function with f_0^λ is generally more interesting than composing a function with $f_{CCM,1}^k$.

5.2 Strengthening the column generation model for the PMP with DFF

To evaluate the DFF in terms of the strength of the corresponding inequalities they generate, we performed a set of computational experiments on random PMP instances. In particular, we compare the results obtained using the method of Vanderbeck described in [28] against the other DFF analyzed in this paper.

All the valid inequalities are derived from the constraints of a column generation model introduced by Vanderbeck in [28], which states as follows.

$$\min \sum_{p \in P} \sum_{n=1}^{ub(P_p)} \lambda_{pn} \quad (8)$$

$$\text{s.t.} \quad \sum_{p \in P} \sum_{n=1}^{ub(P_p)} na_{ip} \lambda_{pn} = b_i, \quad i = 1, \dots, m, \quad (9)$$

$$\sum_{p \in P} \sum_{n=1}^{ub(P_p)} n \lambda_{pn} \leq z_{CSP}, \quad (10)$$

$$\lambda_{pn} \in \{0, 1\}, \quad p \in P, \quad n = 1, \dots, ub(P_p). \quad (11)$$

The whole set of patterns is denoted by P , with P_p being the p^{th} pattern in P . In this model, each column represents a pattern P_p replicated n times (n stands for the multiplicity of the pattern). The binary decision variables λ_{pn} indicate if a pattern P_p with multiplicity n is used or not. The constraints of the model are related to the demand b_i for each item i , and to the number of rolls that can be used (z_{CSP} is usually equal to the minimum number of rolls required to cut all the items). Finally, $ub(P_p)$ is the maximum multiplicity allowed for pattern P_p , and it is equal to $\min_{i=1, \dots, m} \left\lfloor \frac{b_i}{a_{ip}} \right\rfloor$. Coefficients a_{ip} represent the number of items i in pattern P_p .

The bound of this model is very weak. Hence, Vanderbeck [28] proposed to use the set of superadditive functions analyzed at the beginning of Section 4.4. He proved that the resulting valid inequalities were stronger than the rank 1 Chvátal-Gomory cuts. However, as shown in [2], they are not maximal. One of the advantages of using DFF for the PMP (and other problems) is that they can generate good cuts

even when applied directly to the original coefficients of the column generation model. This almost guarantees that the cutting plane generation procedure will not induce any major complication to the pricing subproblems. Furthermore, the separation procedures are generally fast. In [2], some very simple linear combinations were tried producing a real improvement on the linear bound. Indeed, when we know how a specific DFF works, it is usually not difficult to anticipate a linear combination that is likely to produce better cuts. Broadly speaking, the larger the left-hand side coefficients and the smaller the right-hand side coefficient of a particular constraint, the stronger will be the cut generated by many DFF from it. For example, instead of generating cuts from (10), we can derive cuts from

$$\sum_{p \in P} \sum_{n=1}^{ub(P_p)} (n-1)\lambda_{pn} \leq z_{CSP} - lb_{PMP}, \quad (12)$$

where lb_{PMP} stands for the best known lower bound for a particular instance of the PMP. Constraint (12) consists in the combination of (10) with the following trivial constraint:

$$\sum_{p \in P} \sum_{n=1}^{ub(P_p)} \lambda_{pn} \geq lb_{PMP}.$$

Following this principle, one can generate valid inequalities from DFF using the most appropriate constraints. For the PMP, cuts can be derived for example using the following valid waste constraint:

$$\sum_{p \in P} \sum_{n=1}^{ub(P_p)} (W - \sum_{i=1}^m w_i a_{ip}) \lambda_{pn} \leq W z_{CSP} - \sum_{i=1}^m w_i b_i. \quad (13)$$

where w_i and W are the sizes of the items and the rolls, respectively.

For our experiments, we used a total of 500 instances divided in 10 groups of 50 instances. These instances were generated using different numbers of items (n) and different intervals for the sizes of the items ($[min, 800]$). The size of the roll (W) is equal to 1000. The instances were generated using the CUTGEN1 generator [19]. The following table describes each set of instances.

Set	min	W	n
A-1	300	1000	200
A-2	300	1000	400
A-3	300	1000	600
A-4	300	1000	1200
B-1	100	1000	200
B-2	100	1000	400
B-3	100	1000	600
B-4	100	1000	1200
C-1	10	1000	200
C-2	10	1000	600

Table 9: Instances of *PMP* used

Tables 10, 11 and 12 summarize respectively the total value of the linear bounds obtained when a specific DFF is used to generate valid inequalities, the number of times a particular DFF led to the best linear bound for each set of instances and the number of times a DFF was the only one to lead to the strongest linear program. In Table 10, the values in column *cg* correspond to the bounds obtained using standard column generation without any additional valid inequality. Again, DFF that are dominated by other DFF are not considered in Table 12. To generate valid cutting planes, the DFF were applied to the demand constraints (9), to the constraint limiting the total number of rolls (10), and to the waste constraint (13).

The first observation is that, whatever the DFF we choose, the column generation bound is always improved. The other dominance results stated along the paper are also confirmed by these results: $f_{CCM,1}^k$ is always better than $f_{FS,2}^k$, while the cuts returned by $f_{LL,2}^k$ dominate those given by $f_{LL,1}^k$. For these

Set	Sum										
	cg	$f_{VB,1}^k$	f_0^λ	$f_{BJ,1}^k$	$f_{LL,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$	$f_{FS,2}^k$	$f_{CCM,1}^k$	$f_{FS,1}^k$	$f_{VB,2}^k$
A-1	737.8	795.4	787.4	791.8	786.0	789.9	792.7	774.9	791.5	797.7	796.3
A-2	1363.0	1455.8	1453.9	1459.7	1450.8	1455.8	1461.6	1427.3	1461.9	1471.5	1469.4
A-3	739.7	831.8	809.6	820.2	817.4	820.3	822.5	778.4	810.5	831.9	832.1
A-4	1372.8	1543.7	1504.4	1526.1	1522.4	1527.1	1532.7	1444.5	1505.7	1546.5	1544.5
B-1	587.5	659.5	656.7	660.8	654.9	659.3	663.0	638.4	664.8	667.6	666.4
B-2	1086.5	1218.6	1205.4	1210.1	1200.0	1206.8	1213.5	1168.6	1212.8	1222.6	1220.1
B-3	608.0	718.5	694.7	704.5	702.4	705.8	710.9	661.1	696.0	720.4	719.2
B-4	1114.3	1306.7	1275.4	1292.4	1287.4	1292.7	1300.2	1203.9	1270.8	1310.6	1308.3
C-1	534.6	604.6	597.7	600.8	595.4	598.9	602.8	580.6	604.7	608.0	606.1
C-2	551.0	650.1	629.0	640.3	638.1	641.5	644.5	600.0	631.5	652.8	651.1

Table 10: Comparing the DFF in terms of total linear bounds

Set	Best										
	$f_{VB,1}^k$	f_0^λ	$f_{BJ,1}^k$	$f_{LL,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$	$f_{FS,2}^k$	$f_{CCM,1}^k$	$f_{FS,1}^k$	$f_{VB,2}^k$	
A-1	18	6	10	4	7	12	2	18	42	20	
A-2	2	1	2	0	2	2	0	12	37	10	
A-3	16	1	4	4	4	5	1	5	44	26	
A-4	4	0	1	0	0	0	0	0	47	10	
B-1	2	2	6	2	2	5	1	19	33	14	
B-2	6	0	3	1	1	3	0	15	34	11	
B-3	7	1	2	0	2	0	0	2	44	12	
B-4	0	2	2	2	3	0	0	0	43	9	
C-1	7	3	5	4	4	6	1	20	33	12	
C-2	5	0	1	0	3	1	0	0	43	11	

Table 11: Summary: number of times each function leads to the strongest model

Set	Only							
	f_0^λ	$f_{BJ,1}^k$	$f_{LL,2}^k$	$f_{DG,1}^k$	$f_{CCM,1}^k$	$f_{FS,1}^k$	$f_{VB,2}^k$	
A-1	0	1	0	0	7	18	0	
A-2	1	0	0	0	11	28	1	
A-3	0	0	0	0	2	21	4	
A-4	0	1	0	0	0	39	2	
B-1	0	1	0	0	13	20	3	
B-2	0	1	0	0	12	23	3	
B-3	1	1	1	0	1	34	2	
B-4	0	0	1	0	0	38	4	
C-1	0	0	0	0	14	24	2	
C-2	0	0	1	0	0	36	6	

Table 12: Summary: number of times each function is the only best

sets of instances, $f_{FS,1}^k$ is the DFF that generate most regularly the strongest cutting planes. The linear bounds obtained with this DFF are clearly better than those given by the dominated $f_{VB,1}^k$, the DFF proposed by Vanderbeck in [28]. For many instances, $f_{CCM,1}^k$ is the only one to lead to the strongest cuts, and in a non-negligible number of times, the best bound is obtained when the new $f_{VB,2}^k$ is applied.

6 Concluding remarks

In this paper, we surveyed different classical dual-feasible functions used explicitly or not in the literature. A broad characterization and a computational comparison of these functions were performed. We

introduced frameworks that allow to derive and analyze DFF. Several functions hidden in cutting plane generation procedures were formally defined and analyzed. Different properties and relations were stated like maximality and dominance. New dominant functions were proposed. Better bounds and stronger valid inequalities were obtained with these new functions, and with compositions of these functions.

From this paper, it transpires that using functions used, explicitly or not, in the integer programming literature leads to improved lower bounds. The opposite methodology can also be used. Stronger valid inequalities for integer programs can be derived using functions that have been defined originally to compute lower bounds. For problems like the PMP, this strategy led to stronger relaxations that in turn led to state-of-the-art linear lower bounds.

Acknowledgments

We would like to thank the anonymous referees for their constructive comments, which led to a clearer presentation of the material.

We would also like to thank Dr. Jürgen Rietz for his valuable comments.

This work was partially supported by the Portuguese Science and Technology Foundation through the postdoctoral grant SFRH/BPD/24139/2005 for François Clautiaux and the research project POS-C / 57203 / EIA / 2004 for Cláudio Alves and José Valério de Carvalho.

References

- [1] K. Aardal and R. Weismantel. *Polyhedral combinatorics*. Wiley, New York, 1997.
- [2] C. Alves and J. M. Valério de Carvalho. A branch-and-price-and-cut algorithm for the pattern minimization problem. *RAIRO Operations Research (in press)*, 2008.
- [3] R. Baldacci and M. Boschetti. A cutting plane approach for the two-dimensional orthogonal non guillotine cutting stock problem. *European Journal of Operational Research*, 183:1136–1149, 2007.
- [4] M. Boschetti. New lower bounds for the three-dimensional finite bin packing problem. *Discrete Applied Mathematics*, 140:241–258, 2004.
- [5] M. Boschetti and A. Mingozzi. The two-dimensional finite bin packing problem. Part I: New lower bounds for the oriented case. *4OR*, 1:27–42, 2003.
- [6] M. Boschetti and A. Mingozzi. The two-dimensional finite bin packing problem. Part II: New lower and upper bounds. *4OR*, 1:135–147, 2003.
- [7] C. A. Burdett and E. L. Johnson. A subadditive approach to solve linear integer programs. *Annals of Discrete Mathematics*, 1:117–144, 1977.
- [8] A. Caprara and M. Monaci. Bidimensional packing by bilinear programming. *Mathematical Programming (in press)*, 2007.
- [9] J. Carlier, F. Clautiaux, and A. Moukrim. New reduction procedures and lower bounds for the two-dimensional bin packing problem with fixed orientation. *Computers and Operations Research*, 34(8):2223–2250, 2007.
- [10] J. Carlier and E. Néron. A new LP-based lower bound for the cumulative scheduling problem. *European Journal of Operational Research*, 127:363–382, 2000.
- [11] J. Carlier and E. Néron. On linear lower bounds for the resource constraint project scheduling problem. *European Journal of Operational Research*, 149(2):314–324, 2003.
- [12] J. Carlier and E. Néron. Computing redundant resources for the resource constrained project scheduling problem. *European Journal of Operational Research*, 176(3):1452–1463, 2007.

- [13] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Math.*, 4:305–337, 1973.
- [14] F. Clautiaux. *Bornes inférieures et méthodes exactes pour le problème de bin-packing en deux dimensions avec orientation fixe*. PhD thesis, Université de Technologie de Compiègne, France, 2005.
- [15] S. Dash and O. Günlük. Valid inequalities based on simple mixed-integer sets. *Mathematical Programming*, 105:29–53, 2006.
- [16] S. Fekete and J. Schepers. New classes of fast lower bounds for bin packing problems. *Mathematical Programming*, 91:11–31, 2001.
- [17] S. Fekete and J. Schepers. A general framework for bounds for higher-dimensional orthogonal packing problems. *Mathematical Methods of Operations Research*, 60:311–329, 2004.
- [18] M. R. Garey and D. S. Johnson. *Computers and intractability, a guide to the theory of NP-completeness*. Freeman, New York, 1979.
- [19] T. Gau and G. Waescher. CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research*, 84:572–579, 1995.
- [20] P. Gilmore and R. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.
- [21] P. Gilmore and R. Gomory. A linear programming approach to the cutting stock problem - part II. *Operations Research*, 11:863–888, 1963.
- [22] R. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.
- [23] J. B. Lasserre. Integer programming, duality and superadditive functions. *Contemporary Mathematics*, 374:139–150, 2005.
- [24] A. N. Letchford and A. Lodi. Strengthening Chvátal-Gomory cuts and Gomory fractional cuts. *Operations Research Letters*, 30:74–82, 2002.
- [25] G. S. Lueker. Bin packing with items uniformly distributed over intervals [a,b]. In *Proc. of the 24th Annual Symposium on Foundations of Computer Science (FOCS 83)*, pages 289–297. IEEE Computer Society, 1983.
- [26] S. Martello and P. Toth. *Knapsack problems - Algorithms and Computer Implementation*. Wiley, Chichester, 1990.
- [27] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, 1998.
- [28] F. Vanderbeck. Exact algorithm for minimizing the number of setups in the one-dimensional cutting stock problem. *Operations Research*, 46(6):915–926, 2000.
- [29] G. Wäscher, H. Haussner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109–1130, 2007.