ELSEVIER

European Journal of Operational Research 141 (2002) 253–273

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

LP models for bin packing and cutting stock problems

J.M. Valério de Carvalho

Dept. Produção e Sistemas, Universidade do Minho, 4710-057 Braga, Portugal

Received 4 October 2000; accepted 18 September 2001

Abstract

We review several linear programming (LP) formulations for the one-dimensional cutting stock and bin packing problems, namely, the models of Kantorovich, Gilmore–Gomory, onecut models, as in the Dyckhoff–Stadtler approach, position-indexed models, and a model derived from the vehicle routing literature.

We analyse some relations between the corresponding LP relaxations, and their relative strengths, and refer how to derive branching schemes that can be used in the exact solution of these problems, using branch-and-price. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Bin packing; Cutting stock

1. Introduction

The classic one-dimensional cutting stock problem consists of determining the smallest number of rolls of width W that have to be cut in order to satisfy the demand of m clients with orders of b_i rolls of width w_i, $i = 1, 2, \dots, m$. According to Dyckhoff's system [14], this problem is classified as 1/V/I/R, which means that it is a one-dimensional problem with an unlimited supply of rolls of identical size and a set of orders that must be fulfilled. The last entry in the classification system means that the quantities ordered of each item type are large, that is, the average demand per order width, also denoted as the multiplicity factor, is large.

On the other hand, the bin packing problem can be stated as follows: given a positive integer bin of capacity W and m items of integer sizes $w_1, \dots, w_i, \dots, w_m$ $(0 < w_i \leqslant W,\ i = 1, \dots, m)$, the problem is to assign the items to the bins so that the capacity of the bins is not exceeded and the number of bins used is minimized. Under Dyckhoff's system, this problem is classified as 1/V/I/M, meaning that it is a one-dimensional problem with an unlimited supply of bins of identical size and a set of orders and many items of many different sizes, yielding a low multiplicity factor.

The one-dimensional cutting stock problem and bin packing problem are indeed very similar, and the reason that possibly motivated the use of a different classification for the problems was that different solution methods had been traditionally used to address them.

E-mail address: vc@dps.uminho.pt (J.M. Valério de Carvalho).

0377-2217/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0377-2217(02)00124-8

Linear programming (LP) was used to address the cutting stock problem, where the average demand per order width is large, because integer solutions of very good quality can usually be obtained from the solution of the LP relaxation using heuristics. Waescher and Gau [34] did extensive computational experiments with instances with an average demand per order width of 10 and 50, and concluded that the optimal integer solutions could be obtained in most cases. Furthermore, when the heuristics of Waescher and Gau [34] and the heuristic of Stadtler [26] are used in conjunction with each other, they solve almost every instance of the cutting stock problem to an optimum.

However, if the average demand per order width is very low, as it occurs in the bin packing problem, where the multiplicity factor may be close to one or even equal to one, the decision variables in the optimal solution of the LP relaxation have often values that are a fraction of unity, and it may not be so easy for heuristics to find the integer optimal solution. Therefore, to obtain integer solutions to bin packing problems, other techniques were and are usually used, as, for example, in [23,27,28].

LP is a powerful tool to address integer programming and combinatorial optimization problems, but, in the past, it was recognized that it was not easy to use it to find exact solutions to the integer cutting stock problem, because it is necessary to combine column generation with tools to obtain integer solutions to the cutting stock problem. We quote the final comments of a 1979 paper by Gilmore [17]:

> "A linear programming formulation of a cutting stock problem results in a matrix with many columns. A linear programming formulation of an integer programming problem results in a matrix with many rows. Clearly a linear programming formulation of an integer cutting stock problem results in a matrix that has many columns and many rows. It is not surprising that it is difficult to find exact solutions to the integer cutting stock problem."

Nevertheless, Vance [29,30], Vanderbeck [31–33] and Valério de Carvalho [4] recently presented some attempts at combining column generation and branch-and-bound, a framework that has also been successfully applied to other integer programming problems, and is usually denoted as branch-and-price. They were able to solve exactly quite large instances of bin packing and cutting stock problems.

In particular, Valério de Carvalho consistently solved all the bin packing instances from the OR-Library, while other tools failed to solve some of the instances (see [4] for details). The bin packing instances of the OR-Library can be seen as cutting stock instances with only a few items for each demand width. For example, the instances in the $t-501$ class have about 200 different demand widths and a total of 501 items, which yields a multiplicity factor of about 2.5. The experiments show that solving strong LP models with branch-and-price is a useful framework for obtaining integer solutions for both cutting stock (1/V/I/R) and bin packing (1/V/I/M) problems.

In this paper, we review some LP formulations for the one-dimensional cutting stock and bin packing problems, both for problems with identical and non-identical large objects, mainly following the material presented in the original papers. In the cases where models are presented for problems with identical objects, it is not very difficult to see how those models could be extended to cater for the problems with non-identical large objects.

We try to highlight the relations between the models, and their decision variables. The interest in the decision variables is mainly motivated by their possible use in the design of new branching schemes for branch-and-price algorithms for solving cutting stock and bin packing problems with rolls of identical or multiple sizes, possibly with other constraints, as knife constraints. The decision variables in the LP models may be possible candidates for partitioning, yielding finite branching schemes with guaranteed convergence, that avoid symmetry and preserve the structure of the subproblem (for a discussion of these issues, see [2]).

We will use indistinctly the terminology applied either in the bin packing problem or in the cutting stock problem, when referring to the large objects, the small objects or the problems. For instance, we address the model where the large objects may be non-identical. This problem can be denoted as the variable sized bin packing problem or as the multiple widths (or lengths) cutting stock problem.

In Section 2, we present the cutting stock model introduced by Kantorovich, and comment on the quality of the bounds that result from the LP relaxation of the model. In Section 3, we present some concepts from the Dantzig–Wolfe decomposition. In Section 4, it is showed how the Gilmore–Gomory model can be obtained applying a Dantzig–Wolfe decomposition to the Kantorovich model. In Section 5, we present two models based on position-indexed formulations, one based on arc flows, while the other has consecutive ones. Their equivalence to Gilmore–Gomory models is also shown. In Section 6, we present onecut models, introduced by Dyckhoff and Stadtler. In Section 7, we refer to an extended model that combines columns from the Gilmore–Gomory model and other columns that were denoted as dual cuts. In Section 8, the bin packing problem is seen as a special case of a vehicle routing problem. Finally, in Section 9, we present some conclusions.

2. Kantorovich model

Kantorovich [21] introduced the following mathematical programming formulation for the cutting stock problem to minimize the number of rolls used to cut all the items:

$$\min \quad \sum_{k=1}^{K} y_k \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} x_{ik} \geqslant b_i, \quad i = 1, \ldots, m, \tag{2}$$

$$\sum_{i=1}^{n} w_i x_{ik} \leqslant W y_k, \quad k = 1, \ldots, K, \tag{3}$$

$$y_k = 0 \text{ or } 1, \quad k = 1, \ldots, K, \tag{4}$$

$$x_{ik} \geqslant 0 \text{ and integer}, \quad i = 1, \ldots, m, \; k = 1, \ldots, K, \tag{5}$$

where K is a known upper bound on the number of rolls needed, $y_k = 1$, if roll k is used, and 0, otherwise, and x_{ik} is the number of times item i is cut in roll k.

A lower bound for the optimum can be obtained from the optimum of its LP relaxation, which results from substituting the two last constraints for $0 \leqslant y_k \leqslant 1$ and $x_{ik} \geqslant 0$.

Martello and Toth [23] showed that the lower bound provided by the LP relaxation can be very weak. Actually, they proved it for the bin packing problem, but the result can be easily extended to the cutting stock problem. In the bin packing problem, where items are considered separately, and the x_{ik} are required to be binary, the LP bound is equal to $\lceil \sum_{i=1}^{m} w_i / W \rceil$.

This bound is equal to the minimum amount of space that is necessary to accommodate all the items, and can be very poor for instances with large waste. In the limit, as W increases, when all the items have a size $w_i = \lfloor W/2 + 1 \rfloor$, the lower bound approaches 1/2.

This is a drawback of the model. Good quality lower bounds are of vital importance when using LP based approaches to solve integer problems. Valério de Carvalho [4] and Vance [29] report that branch-and-bound algorithms based on this model failed to solve to optimality some instances of the bin packing and the cutting stock problems, respectively, that could otherwise be tackled using stronger formulations.

3. Dantzig–Wolfe decomposition

The Dantzig–Wolfe decomposition is a powerful tool that can be used to obtain models for integer and combinatorial optimization problems with stronger LP relaxations. After reviewing the basic concepts, we show its application to the bin packing and cutting stock problems.

Many integer programming models have a nice structure and a constraint set that can be partitioned and expressed as follows:

$$\min \quad cx \tag{6}$$

$$\text{s.t.} \quad Ax = b, \tag{7}$$

$$x \in X, \tag{8}$$

$$x \geqslant 0 \text{ and integer.} \tag{9}$$

The LP relaxation of this model, that results from dropping the integrality constraints on the variables x can be very weak. An illustrative example has just been shown. A stronger model can be obtained by restricting the set of points from set X that are considered in the LP relaxation of the reformulated model.

According to Minkowski's Theorem (see [24]), any point x of an non-empty polyhedron X can be expressed as a convex combination of the extreme points of X plus a non-negative linear combination of the extreme rays of X,

$$X = \left\{ x \in \mathbb{R}^n_+ : \ x = \sum_{p \in P} \lambda^p x^p + \sum_{r \in R} \mu^r r^r, \ \sum_{p \in P} \lambda^p = 1, \ \lambda^p \geqslant 0 \ \forall p \in P, \ \mu^r \geqslant 0 \ \forall r \in R \right\},$$

where $\{x^p\}_{p \in P}$ is the set of extreme points of X and $\{r^r\}_{r \in R}$ is the set of extreme rays of X.

Substituting the value of x in the original model, we obtain the following LP model:

$$\min \quad c\left(\sum_{p \in P} \lambda^p x^p + \sum_{r \in R} \mu^r r^r \right) \tag{10}$$

$$\text{s.t.} \quad A\left(\sum_{p \in P} \lambda^p x^p + \sum_{r \in R} \mu^r r^r \right) = b, \tag{11}$$

$$\sum_{p \in P} \lambda^p = 1, \tag{12}$$

$$\lambda^p \geqslant 0 \quad \forall p \in P, \tag{13}$$

$$\mu^r \geqslant 0 \quad \forall r \in R. \tag{14}$$

Eq. (12) is usually denoted as a convexity constraint. After a rearrangement of the terms, we have

$$\min \quad \sum_{p \in P} (cx^p)\lambda^p + \sum_{r \in R} (cr^r)\mu^r \tag{15}$$

$$\text{s.t.} \quad \sum_{p \in P} (Ax^p)\lambda^p + \sum_{r \in R} (Ar^r)\mu^r = b, \tag{16}$$

$$\sum_{p \in P} \lambda^p = 1, \tag{17}$$

$$\lambda^p \geqslant 0 \quad \forall p \in P, \tag{18}$$

$$\mu^r \geqslant 0 \quad \forall r \in R. \tag{19}$$

In the Dantzig–Wolfe decomposition, this problem is usually denoted as the master problem. Its decision variables are $\lambda^p\ \forall p \in P$, and $\mu^r\ \forall r \in R$, and its columns correspond to the extreme points and the extreme rays of X. The number of columns in the master problem may be huge, and it is unpractical to enumerate them all, and solve the corresponding LP model. Therefore, this problem is usually solved by column generation.

The dual information given by the master problem is used to price the columns that are out of the master problem, and can potentially improve the objective function. The most attractive column can be found by solving a subproblem [22]. Usually, the subproblem is a well structured optimization problem.

The attractive columns that are successively inserted in the master problem correspond to optimal solutions of the subproblem, and to extreme points and extreme rays of set X. When all the extreme points and extreme rays of X are integer, a situation in which we say that the subproblem has the integrality property, the LP relaxation of the original model and the relaxation of the reformulated model that results from the Dantzig–Wolfe decomposition have the same LP bounds [16].

However, when the subproblem does not have the integrality property, the relaxation of the reformulated model will provide a better LP bound. That happens in the cutting stock problem, where the subproblem is a knapsack problem, as follows.

4. A convexity model

Vance [29] applied a Dantzig–Wolfe decomposition to Kantorovich's model, keeping constraints (2) in the master problem, and being the subproblem defined by the integer solutions to the knapsack constraints (3).

Kantorovich's model has several knapsack constraints, one for each roll k. Each knapsack constraint defines a bounded set, having only extreme points, and no extreme rays. The set X is the intersection of the knapsack constraints for each roll. The solution set of the knapsack constraint of roll k is a polytope that can have fractional extreme points. If we determine the optimal integer solution of the knapsack subproblems, we eliminate the valid fractional solutions of the knapsack constraints that do not correspond to valid cutting patterns for roll k.

Only integer (and maximal) solutions to the knapsack constraint for roll k, are considered, that is, those such that $\sum_{i=1}^m w_i a_{ik} \leqslant W$, $a_{ik} \geqslant 0$, and integer. These solutions are described by the vector $(a_{1k}^p, \ldots, a_{ik}^p, \ldots, a_{mk}^p)^{\mathrm{T}}\ \forall p \in P$, where P is the set of indexes of the valid cutting patterns.

The reformulated model is as follows:

$$\min \quad \sum_{k=1}^{K} \sum_{p \in P} \lambda_k^p \tag{20}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \sum_{p \in P} a_{ik}^p \lambda_k^p \geqslant b_i, \quad i = 1, \ldots, m, \tag{21}$$

$$\sum_{p \in P} \lambda_k^p \leqslant 1, \quad k = 1, \ldots, K, \tag{22}$$

$$\sum_{p \in P} a_{ik}^p \lambda_k^p \geqslant 0, \text{ and integer}, \quad i = 1, \ldots, m, \ k = 1, \ldots, K, \tag{23}$$

$$\sum_{p \in P} \lambda_k^p \geqslant 0, \text{ and integer}, \quad k = 1, \ldots, K, \tag{24}$$

$$\lambda_k^p \geqslant 0 \quad \forall p \in P, \ k = 1, \ldots, K. \tag{25}$$

The weights of the convex combination of the valid cutting patterns, λ_k^p, are the decision variables in the reformulated model, and have a very precise meaning: λ_k^p is the number of times pattern p is produced in roll

k. The convexity constraint can be expressed as a less than or equal to constraint, because the null solution to the knapsack subproblem k, $(0,\ldots,0)^{\mathrm{T}} \in \mathbb{N}^m$, is also a valid extreme point. The corresponding column in the reformulated model has null coefficients in constraints (21) and a coefficient 1 in the convexity constraint for roll k, and therefore has the structure of a slack variable. When it takes the value 1, it means that roll k is not cut.

Constraints (23) are implied by constraints (24) and by the fact that all $a_{ik}^p \geqslant 0$ and integer. Nevertheless, they are explicitly expressed, because they were used in [29] to derive a branching scheme for this model. See also [6] for a note concerning this branching scheme.

The LP relaxation of the reformulated model provides a stronger lower bound for the cutting stock problem, because the only feasible solutions are those that are non-negative linear combinations of the integer solutions to the knapsack problem. Non-negative linear combinations of the fractional extreme points of the knapsack polytope, which are feasible in the LP relaxation of the model introduced by Kantorovich, are thus eliminated.

The subproblem decomposes into $|K|$ subproblems, which are knapsack problems. Vance [29] showed that when all the rolls have the same width, being all the subproblems identical, the reformulated model is equivalent to the classical Gilmore–Gomory model. The possible cuttings pattern are described by the vector $(a_1^p,\ldots,a_i^p,\ldots,a_m^p)^{\mathrm{T}}$, where the element a_i^p represents the number of rolls of width w_i obtained in cutting pattern p. Let λ^p be a decision variable that designates the number of rolls to be cut according to cutting pattern p.

The cutting stock problem is thus modelled as follows:

$$\min \quad \sum_{p\in P} \lambda^p \tag{26}$$

$$\text{s.t.} \quad \sum_{p\in P} a_i^p \lambda^p \geqslant b_i, \quad i = 1,2,\ldots,m, \tag{27}$$

$$\lambda^p \geqslant 0 \text{ and integer} \quad \forall p \in P, \tag{28}$$

and for the cutting pattern to be valid:

$$\sum_{i=1}^{m} a_i^p w_i \leqslant W, \tag{29}$$

$$a_i^p \geqslant 0 \text{ and integer} \quad \forall p \in P. \tag{30}$$

The number of columns in formulation (26)–(28) is exponential, and, even for moderately sized problems, may be very large. As it is impractical to enumerate all the columns to solve the linear relaxation of this problem, Gilmore and Gomory [19] introduced column generation.

Vance developed a branch-and-price algorithm for solving this convexity model, using a partition rule derived by Johnson [20] for problems with convexity rows. When all the roll sizes are identical, undesirable symmetry problems arise. There is symmetry when there are solutions that, being different in terms of the values of the decision variables, correspond to the same cutting solution. Symmetry may be detrimental if those solutions are explored in different nodes of the branch-and-bound tree, leading to wasted computational time.

5. Position-indexed models

In position-indexed formulations, variables that correspond to items of a given type are indexed by the physical position they occupy inside the large objects. This type of formulations was used by Beasley [3] in a model for the solution of a two-dimensional non-guillotine cutting stock problem, and also in scheduling [7,8], where they were coined as 'time-indexed formulations'.

In the one-dimensional cutting stock problem, in a position-indexed formulation, a variable represents the placement of an item at a given distance from the border of the roll. That happens in the arc flow formulation and in the model with consecutive ones that are presented next. We address the formulations for the variable sized bin packing problem, which are an extension of the model introduced in [4] for the problem with identical bins.

5.1. Arc flow model

In the variable sized bin packing problem, we are given a finite set of bins of integer non-identical capacities and a set of items of integer size to pack in the bins. We consider the problem of packing all the items to minimize the amount of capacity used. This objective is equivalent to minimizing the wasted material that results from not filling entirely the used bins, when we seek a solution with the exact quantities produced.

Given a list of bins, the bins are grouped into K classes of different bin capacities W_k, $k = 1, \ldots, K$, being B_k, $k = 1, \ldots, K$, the number of bins in each class. The items are also grouped into m classes of different item sizes w_i, $i = 1, \ldots, m$, being b_i, $i = 1, \ldots, m$, the number of items in each class. After grouping, we assume that the classes of bins and the classes of items are indexed in order of decreasing values of capacity and size, respectively.

Let $W_{\max} = \max_k W_k = W_1$. Consider a graph $G = (V, A)$ with a set of vertices $V = \{0, 1, 2, \ldots, W_{\max}\}$ and a set of arcs $A = \{(d, e) : 0 \leqslant d < e \leqslant W_{\max}$ and $e - d = w_i$ for every $1 \leqslant i \leqslant m\}$, meaning that there exists a directed arc between two vertices if there is an item of the corresponding size. Consider also additional arcs between $(d, d+1)$, $d = 0, 1, \ldots, W_{\max} - 1$ corresponding to unoccupied portions of the bin. The number of arcs is $O(mW_{\max})$.

There is a packing in a single bin of capacity W_k iff there is a path between vertices 0 and W_k. The length of arcs that constitute the path define the item sizes to be packed.

In the same set of vertices, consider directed arcs from vertex W_k to vertex 0, if there is a bin of capacity W_k, $k = 1, \ldots, K$. The number of bin arcs is K, which is $O(W_{\max})$, and, thus, the total number of arcs is $O(mW_{\max})$. The number of constraints in the model is $O(W_{\max} + m + K)$.

5.1.1. Reduction criteria

Using these variables, there are many alternative solutions with exactly the same items in each bin. We may reduce both the symmetry of the solution space and the size of the model by considering only a subset of arcs from A. If we search a solution in which the items are ordered in decreasing values of size, the following criteria, presented in [4], may be used to reduce the number of arcs that are taken into account.

Criterion 1. Let w_{i_1} and w_{i_2} denote the sizes of any two items such that $w_{i_1} \geqslant w_{i_2}$. An arc of size w_{i_2}, designated by $(d, d + w_{i_2})$, can only have its tail at a node d that is the head of another arc of size w_{i_1}, $(d - w_{i_1}, d)$, or, else, from node 0, i.e., the left border of the bin.

In particular, if a bin has any loss, it will appear last in the bin. A bin can never start with loss.

Criterion 2. All the loss arc variables $x_{d,d+1}$ can be set to zero for $d < w_m$.

In a bin, the number of consecutive arcs corresponding to a single item size must be less than or equal to the number of items of that size. Therefore, following Criterion 1.

Criterion 3. Let i_1 and i_2 denote any two item sizes such that $w_{i_1} > w_{i_2}$. Given any node d that is the head of another arc of size w_{i_1} or $d = 0$, the only valid arcs for size w_{i_2} are those that start at nodes $d + sw_{i_2}$, $s = 0, 1, \ldots, b_{i_2} - 1$ and $d + sw_{i_2} \leqslant W_{\max}$, where b_{i_2} is the demand of items of size w_{i_2}.

Let $A' \subset A$ be the set of arcs that remain after applying the above criteria. A model for the variable sized bin packing problem is presented in the following example:

Example 5.1. We use an example from Dyckhoff [12]. Consider a set of bins of widths 9, 6 and 5, available in quantities B_1, B_2 and B_3, respectively. The bins are grouped in three classes, with capacities and availability of $\mathscr{W} = (9, 6, 5)$ and $\mathscr{B} = (B_1, B_2, B_3)$, respectively. The items are also grouped in three classes, with sizes and demands of $w = (4, 3, 2)$ and $b = (20, 10, 20)$, respectively. The available capacity is equal to $\sum_k W_k B_k = 9B_1 + 6B_2 + 5B_3$, while the sum of sizes of the items is $\sum_i w_i b_i = 150$. After applying the reduction criteria presented above, the set of arcs is the one presented in Fig. 1.

5.1.2. Mathematical formulation

Consider decision variables x_{de}, associated with the item arcs defined above, which correspond to the number of items of size $e - d$ placed in any bin at the distance of d units from the beginning of the bin. Consider also decision variables z_k, $k = 1, \ldots, K$, associated with bin arcs, that correspond to the number of bins of capacity W_k used. The variable z_k can be seen as a feedback arc, from vertex W_k to vertex 0, and could also be denoted as $x_{W_k,0}$.

This problem is formulated as the problem of minimizing the sum of the capacities of the bins that are necessary to pack all the items. This objective corresponds to finding the minimum weight flow (the weights are the capacities of the bins), subject to the constraint that the sum of the flows in the arcs of each item size is greater than or equal to the corresponding demand. The model is as follows:

$$\min \quad \sum_{k=1}^{K} W_k z_k \tag{31}$$

$$\text{s.t.} \quad -\sum_{(d,e)\in A'} x_{de} + \sum_{(e,f)\in A'} x_{ef} = \begin{cases} \sum_{k=1}^{K} z_k & \text{if } e = 0, \\ -z_k & \text{for } e = W_k, \quad k = 1, \ldots, K, \\ 0 & \text{otherwise,} \end{cases} \tag{32}$$

$$\sum_{(d,d+w_i)\in A'} x_{d,d+w_i} \geqslant b_i, \quad i = 1, \ldots, m, \tag{33}$$

$$z_k \leqslant B_k, \quad k = 1, \ldots, K, \tag{34}$$

$$x_{de} \geqslant 0 \text{ and integer} \quad \forall (d,e) \in A', \tag{35}$$

$$z_k \geqslant 0 \text{ and integer}, \quad k = 1, \ldots, K. \tag{36}$$

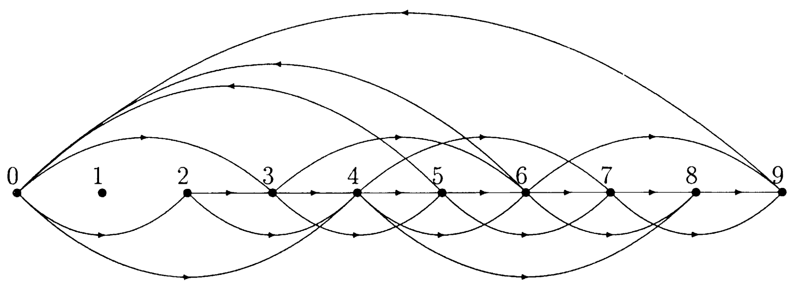

Fig. 1. Graph for arc flow model.

Constraints (32) are flow conservation constraints. They ensure that the flows correspond to a valid packing, because an item is either placed at the border of the bin or immediately after another item. Constraints (33) enforce that the demand is satisfied, while constraints (34) guarantee that the amount of flow in the feedback arc of a given bin capacity is limited to the available number of bins of that capacity. Notice that, under these availability constraints, the problem may be infeasible.

By the flow decomposition properties [1], non-negative flows can be decomposed into a finite set of paths and cycles. The set of constraints (32) defines a homogeneous system, and, therefore, there are no excess and no deficit nodes. A given flow decomposes into a set of cycles, and each cycle has item arcs and a single feedback arc, that corresponds to a given bin size.

Example 5.2. The LP model of Example 5.1 is presented in Fig. 2.

This model has some symmetry, because there may be different paths that correspond to the same cutting pattern. In instances with a small average number of items per bin, which happen to be rather difficult instances, if the criteria mentioned above are applied, there is low symmetry, and its undesirable effects are not so harmful.

This model is interesting mainly because it provides a branching scheme for a branch-and-price algorithm that does not destroy the structure of the subproblem, which can be seen as a longest path problem in an acyclic digraph, and solved using dynamic programming. For the cutting stock problem with identical roll sizes, Valério de Carvalho [4] presents a branching scheme that is based on a similar arc flow model, and the branching constraints are placed in the arc flows.

Afterwards, we show that applying a Dantzig–Wolfe decomposition to the arc flow formulation (used to model the variable sized bin packing problem), we obtain the model for the multiple lengths cutting stock problem that was denoted in Gilmore and Gomory [19] as the machine balance problem; this model will be presented in the sequel.

Proposition 5.1. *The LP arc flow model* (31)–(34) *is equivalent to the machine balance problem of Gilmore and Gomory.*

	z_1	z_2	z_3	x_{04}	x_{48}	x_{03}	x_{36}	x_{47}	x_{69}	x_{02}	x_{24}	x_{35}	x_{46}	x_{57}	x_{68}	x_{79}	x_{23}	x_{34}	x_{45}	x_{56}	x_{67}	x_{78}	x_{89}	
node 0	1	1	1	-1		-1				-1														$= 0$
1																								$= 0$
2										1	-1						-1							$= 0$
3						1	-1					-1					1	-1						$= 0$
4				1	-1			-1			1		-1					1	-1					$= 0$
5			-1									1		-1					1	-1				$= 0$
6		-1					1		-1				1		-1					1	-1			$= 0$
7								1						1		-1					1	-1		$= 0$
8					1										1							1	-1	$= 0$
9	-1								1							1							1	$= 0$
$w_i = 4$				1	1																			≥ 20
3						1	1	1	1															≥ 10
2										1	1	1	1	1	1	1								≥ 20
$W_k = 9$	1																							$\leq B_1$
6		1																						$\leq B_2$
5			1																					$\leq B_3$

Fig. 2. Arc flow model.

Proof. We can obtain an equivalent formulation for the variable sized bin packing problem applying a Dantzig–Wolfe decomposition to the LP relaxation of the arc flow model, keeping constraints (33) and (34) in the master problem, and letting constraints (32) define the subproblem.

The set of constraints (32) and the non-negativity constraints without the integrality requirements define a homogeneous system, that corresponds to a set X. This system has only one extreme point, the solution with null flow, and all other valid flows can be expressed as a non-negative linear combinations of circulation flows along cycles. Each cycle will correspond to a valid packing, and is defined by a unique bin and a set of items. The cycles start at node 0, include a set of item arcs, and eventually loss arcs, and return to node 0, through a feedback arc, that corresponds to a bin.

The circulation flows along each cycle cannot be expressed as non-negative linear combinations of other circulation flows, and are, therefore, extremal. The extremal flows are not bounded and each cycle will correspond to an extreme ray. Therefore, the reformulated problem will not have a convexity constraint.

The subproblem will only generate extreme rays to the master problem. Let Γ be the set of feasible cycles. For each different capacity W_k, there will be a set of valid packing solutions. Let Γ^k be the set of feasible cycles for bin k, $k = 1, \ldots, K$. The sets Γ_k are mutually disjoint and $\Gamma = \bigcup_k \Gamma^k$.

Each cycle $r \in \Gamma^k$ can be described using the binary variables x^r_{de} and z^r_k, that take the value 1, if the corresponding arc is included in the cycle. A column in the master problem can be defined by $(\tilde{a}^r_k, \tilde{b}^r_k)$, where $\tilde{a}^r_k = (a^r_{1k}, \ldots, a^r_{ik}, \ldots, a^r_{mk})$ is the vector that defines the number of items for each order and $\tilde{b}^r_k = \tilde{e}_k = (0, \ldots, 1, \ldots, 0) \in \mathbb{N}^K$ is the kth unit vector, with a 1 in position k, that identifies the bin where the items are packed. The coefficients of these columns, a^r_{ik}, are expressed in terms of the decision variables of the subproblem, x^r_{de}, that correspond to the arcs (d, e) that take the value 1 in the shortest path subproblem between nodes 0 and W_k:

$$a^r_{ik} = \sum_{(d,e):e-d=w_i} x^r_{de}, \quad i = 1, \ldots, m, \tag{37}$$

while the element of the vector $\tilde{b}^r_k$ that is equal to 1 is the one that matches z^r_k.

Let μ^r_k be the variables of the master problem, which mean the number of times the packing r is made in bin k. The substitution of these patterns in (31), (33) and (34) gives the following equivalent model:

$$\min \quad \sum_{k=1}^{K} \sum_{r \in \Gamma^k} W_k \mu^r_k \tag{38}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \sum_{r \in \Gamma^k} a^r_{ik} \mu^r_k \geqslant b_i, \quad i = 1, \ldots, m, \tag{39}$$

$$\sum_{r \in \Gamma^k} \mu^r_k \leqslant B_k, \quad k = 1, \ldots, K, \tag{40}$$

$$\mu^r_k \geqslant 0 \text{ and integer}, \quad r \in \Gamma^k, \ k = 1, \ldots, K. \tag{41}$$

This is the model that Gilmore and Gomory denoted as the machine balance model. □

Example 5.3. Consider the instance presented in Example 5.1. The machine balance problem of Gilmore and Gomory is given in Fig. 3. Only maximal patterns, where the amount of loss is smaller than the width of the smallest roll, are presented.

The cycle defined by the arcs $(0, 4)$, $(4, 6)$, $(6, 8)$, $(8, 9)$ and $(9, 0)$ corresponds to the column of the decision variable μ^3_9 of the machine balance model.

When all the rolls have identical size, i.e., $W_k = W \ \forall k$, the model reduces to the classical model of Gilmore and Gomory. The bound given by the linear relaxation of this model is known to be very tight.

	μ_1^1	μ_1^2	μ_1^3	μ_1^4	μ_1^5	μ_1^6	μ_1^7	μ_2^1	μ_2^2	μ_2^3	μ_2^4	μ_3^1	μ_3^2	μ_3^3	
$w_i = 4$	2	1	1					1				1			≥ 20
3		1		3	2	1			2	1			1		≥ 10
2		1	2		1	3	4	1		1	3		1	2	≥ 20
$W_k = 9$	1	1	1	1	1	1	1								$\leq B_1$
6								1	1	1	1				$\leq B_2$
5												1	1	1	$\leq B_3$

Fig. 3. Gilmore–Gomory machine balance model.

5.2. A model with consecutive ones

Applying a unimodular transformation to the arc flow model, that basically sums each flow conservation constraint with the previous one, except for the first, we obtain a formulation with consecutive ones. A column for a variable x_{de} means that, if an item is placed at a distance of d units from the border of the roll, the positions $d, d+1, \ldots, e$ will be physically occupied, and the column that defines the corresponding variable will have a 1 in all those positions, and a 0 otherwise:

$$x_{de} = \begin{cases} 1 & \text{if a piece of width } e-d \text{ is placed at a distance of } d \text{ units from the border,} \\ 0 & \text{otherwise.} \end{cases}$$

The coefficients of the upper part of the matrix that defines the problem are:

$$a_{de}^{r} = \begin{cases} 1 & \text{if position } r \text{ is occupied when a piece of width } e-d \text{ is placed at a} \\ & \text{distance of } d \text{ units from the border,} \\ 0 & \text{otherwise.} \end{cases}$$

Example 5.4. The model that results from applying a unimodular transformation to the arc flow model is presented in Fig. 4.

	z_1	z_2	z_3	x_{04}	x_{48}	x_{03}	x_{36}	x_{47}	x_{69}	x_{02}	x_{24}	x_{35}	x_{46}	x_{57}	x_{68}	x_{79}	x_{23}	x_{34}	x_{45}	x_{56}	x_{67}	x_{78}	x_{89}		
pos. 0-1	1	1	1	-1		-1				-1														=	0
1-2	1	1	1	-1		-1				-1														=	0
2-3	1	1	1	-1		-1					-1						-1							=	0
3-4	1	1	1	-1			-1				-1	-1						-1						=	0
4-5	1	1	1		-1		-1	-1				-1	-1						-1					=	0
5-6	1	1	1		-1		-1	-1					-1	-1						-1				=	0
6-7	1	1			-1			-1	-1					-1	-1						-1			=	0
7-8	1				-1				-1						-1	-1						-1		=	0
8-9	1								-1							-1							-1	=	0
$w_i = 4$				1	1																			$\geq$	20
3						1	1	1	1															$\geq$	10
2										1	1	1	1	1	1	1								$\geq$	20
$W_k = 9$	1																							$\leq$	B_1
6		1																						$\leq$	B_2
5			1																					$\leq$	B_3

Fig. 4. A position-indexed model with consecutive ones.

6. Onecut models

In the Gilmore–Gomory model, each decision variable corresponds to a set of cutting operations that are performed on a large object to obtain the small ordered items.

In onecut models, each decision variable corresponds to a single cutting operation performed on a single piece. Given a piece of some size, the piece is divided into two smaller pieces, denoted as the first section and the second section of the onecut, respectively. Every onecut should produce, at least, one piece of an ordered size. The cutting operations can be performed either on stock pieces or intermediate pieces that result from previous cutting operations.

For any cutting pattern of the Gilmore–Gomory model, starting from a large object, it is easy to derive sequences of onecuts that finally produce the desired cutting pattern. It can be used to model either cutting stock problems with either identical or different large objects.

Onecut models were introduced by Dyckhoff [12], who shows that the Gilmore–Gomory model and the onecut model have equivalent sets of feasible integer solutions. The number of variables in onecut models is pseudopolynomial, and does not grow explosively as in the classical approach. That does not mean that the model is amenable to an exact solution by a good integer LP code, due to the symmetry of the solution space.

The onecut model has more symmetry than the Gilmore–Gomory model, in the sense that there may be many different sequences of onecuts that lead to the same cutting pattern, which is undesirable when searching an optimum integer solution, as pointed by Johnson [20]. A scheme that branches on onecut variables will explore the same solution, or sets of solutions, in many different nodes of the branch-and-bound tree. To our knowledge, the integer solution of onecut models using this, or other branching schemes, has never been tried.

We will present two versions of the onecut model, by Dyckhoff and Stadtler, respectively, in which we include availability constraints that do not appear explicitly in the original papers, where it is assumed that there is an infinite number of rolls of each size available.

6.1. Dyckhoff model

Let S be the set of standard widths, or sizes of large objects $q \in \{W_1, \ldots, W_K\} \subset \mathbb{N}$, and D, the set of order widths $q \in \{w_1, \ldots, w_m\} \subset \mathbb{N}$. It is assumed that $S \cap D = \emptyset$. Let $y_{p,q}$ denote the number of pieces of width p that are divided into a piece of order width q, and a piece of residual width $p - q$. Residual pieces are intermediate pieces in the cutting process, that are not necessarily of an order width, because they can be subsequently cut to obtain a smaller order width, or make trim loss. Sizes of intermediate pieces which are not shorter than the smaller order width make the set of residual widths R. As we are interested in onecuts that split a piece of the set of standard lengths or of the set of residual lengths to obtain, at least, an order width, the valid decisions variables are $y_{p,q}$, $p \in S \cup R$, $q \in D$, $q < p$.

We consider again the same objective function as in the arc flow model. The set of feasible solutions can be formulated as a set of balance constraints, expressed in terms of each width. As before, the decision variable z_k denotes the number of large objects of size W_k used. The model is as follows:

$$\min \quad \sum_{k=1}^{K} W_k z_k \tag{42}$$

$$\text{s.t.} \quad z_k + \sum_{p \in D: p+q \in S \cup R} y_{p+q,p} \geqslant \sum_{p \in D: p<q} y_{q,p} \quad \forall q \in S = \{W_1, \ldots, W_K\}, \tag{43}$$

$$\sum_{p \in S \cup R: p > q} y_{p,q} + \sum_{p \in D: p+q \in S \cup R} y_{p+q,p} \geqslant \sum_{p \in D: p < q} y_{q,p} + N_q \quad \forall q \in (D \cup R) \setminus S, \tag{44}$$

$$z_k \leqslant B_k, \quad k = 1, \ldots, K, \tag{45}$$

$$y_{p,q} \geqslant 0 \text{ and integer}, \quad p \in S \cup R, \ q \in D, \ q < p, \tag{46}$$

$$z_k \geqslant 0 \text{ and integer}, \quad k = 1, \ldots, K, \tag{47}$$

where N_q is the value of the demand of items of size q (for $q \notin D: \ N_q = 0$; for $q \in D$ with $q = w_i: \ N_q = b_i$).

The inequality constraints (43) state that the number of pieces of a given standard width q that can be further split must be less than or equal to the number of large objects of size W_k used, z_k, plus the number of pieces of width q obtained from onecuts with standard or residual pieces of width $p + q$, which are made to obtain demand widths of size p.

For the remaining admissible widths, demand or residual, constraints (44) state that the number of pieces of width q that can be further split or used to satisfy the demand must be less than or equal to the number of pieces of width q obtained from onecuts with larger standard or residual pieces, either when cutting a piece of size p to get an order width q or when cutting a piece of size $p + q$ to get an order width p.

Constraints (45) guarantee that the number of large objects used does not exceed the available number of each size.

Again, let $W_{\max} = \max_k W_k = W_1$. The model has a pseudopolynomial number of variables $O(mW_{\max})$: each onecut has an order width and a residual width. The number of possible residual widths is $O(W_{\max})$; for each width, no more than m different onecut operations can be made. The number of constraints is $O(K + W_{\max})$.

Example 6.1. We use the example from Dyckhoff presented earlier. Here, the set of standard widths is $S = \{9, 6, 5\}$, and the set of order widths is $D = \{4, 3, 2\}$. The quantities ordered are 20, 10 and 20, respectively. From the possible onecut operations, we obtain the following set of residual widths $R = \{7, 6, 5, 4, 3, 2\}$. In Fig. 5, we present the LP formulation for this instance.

6.2. Stadtler model

Stadtler [25] presents an extension of Dyckhoff's model that unveils an interesting structure of the cutting stock problem. Again, each column describes a onecut where two pieces are generated, but the model uses separate constraints for the two pieces generated by the onecut; there are also coupling variables for the order widths T_i, $i = 1, \ldots, m$, to add up the number of pieces of equal width given by the two sections of the cut.

	z_1	z_2	z_3	y_{92}	y_{93}	y_{94}	y_{72}	y_{74}	y_{64}	y_{63}	y_{53}	y_{42}	
9	1			-1	-1	-1							≥ 0
8													≥ 0
7				1			-1	-1					≥ 0
6		1			1				-1	-1			≥ 0
5			1			1	1				-1		≥ 0
4						1		1	1			-1	≥ 20
3					1			1		2	1		≥ 10
2				1			1		1		1	2	≥ 20
$W_k = 9$	1												$\leq B_1$
6		1											$\leq B_2$
5			1										$\leq B_3$

Fig. 5. Dyckhoff model.

	z_1	z_2	z_3	y_{92}	y_{93}	y_{94}	y_{72}	y_{73}	y_{62}	y_{63}	y_{52}	y_{42}	T_1	T_2	T_3	
9	1			-1	-1	-1										$= 0$
8																$= 0$
7				1			-1	-1								$= 0$
6		1			1				-1	-1						$= 0$
5			1			1	1				-1					$= 0$
4								1	1			-1	-1			$= 0$
3										1	1			-1		$= 0$
2												1			-1	$= 0$
1																$= 0$
4						1							1			≥ 20
3					1			1		1				1		≥ 10
2				1			1		1		1	1			1	≥ 20
$W_k = 9$	1															$\leq B_1$
6		1														$\leq B_2$
5			1													$\leq B_3$

Fig. 6. Stadtler model.

Creating different constraints for each piece produced in the onecut, Stadtler gets a model with onecut decision variables with only one -1 and two $+1$ elements. The model has, at most, three non-zero elements per column, and is nicely structured: the set of constraints can be partitioned into two sets, the first with a pure network structure, and the second is composed of generalized upper bounding (GUB) constraints.

Example 6.2. Stadtler's model for the instance presented above is presented in Fig. 6.

There is a remarkable resemblance in the structure of Stadtler model and the arc flow model presented above. The similarity between some variables follows from the fact that placing an item at a given position divides the space to the border of the roll in two portions. At the end of Section 7, we will see that some onecut variables in Dyckhoff model can also be seen as cycles (to be defined later) in the space of the arc flow variables.

7. Extended model

In this section, we consider the cutting stock problem with rolls of identical size, even though the concepts presented here could be extended to the multiple lengths problem. The Gilmore–Gomory model with rolls of identical size is a primal LP problem with an exponential number of columns, $\min\{1 \cdot \lambda : A\lambda = b,\ \lambda \geqslant 0\}$, that is to be solved by column generation, where the columns of A correspond to valid cutting patterns. Its dual is $\max\{\pi b : \pi A \leqslant 1\}$.

The primal solution space is the set of points that are non-negative linear combinations of the valid cutting patterns. Looking at the cutting stock problem from the dual standpoint, the valid dual space is the set of points that obey all dual constraints, that is, the set that would be obtained if all the constraints associated with all valid cutting patterns were considered.

Valério de Carvalho [5] showed that it is possible to obtain an optimal solution to the cutting stock problem by solving an extended model with extra columns. The structure of the new columns is presented below. The were denoted as dual cuts, because they may effectively cut portions of the dual space, as it will be illustrated here by an example, but the optimal dual solutions are preserved, as shown in [5].

Inserting a polynomial number of these dual cuts in the restricted master problem before starting the column generation process, the author experienced a faster convergence of the column generation process, with less columns generated, and a sensible reduction in the number of degenerate pivots, and also in the computational time, when solving the linear relaxation of some very large instances of the cutting stock problem. To see an interpretation for the acceleration, see [5]; here, we focus on the structure of the dual cuts.

If the dual cuts, $\pi D \leqslant d$, are added to the dual problem (and the corresponding columns to the primal problem), we get the following extended primal–dual pair of problems:

$$\begin{array}{llcll} \min & 1 \cdot \lambda + dv & \qquad & \max & \pi b \\ \text{s.t.} & A\lambda + Dv = b, & & \text{s.t.} & \pi A \leqslant 1, \\ & \lambda,\ v \geqslant 0, & & & \pi D \leqslant d. \end{array}$$

The following proposition renders possible an algorithm based on the following idea. First, the extended model is solved to optimality, and then an optimal solution to Gilmore–Gomory model is built.

Proposition 7.1 [5]. *Consider an extended model for the one-dimensional cutting stock problem with the following set of dual cuts*:

$$-\pi_i + \sum_{s \in S} \pi_s \leqslant 0 \quad \forall i, S \tag{48}$$

for any given width w_i, and a corresponding set S of item widths, indexed by s, such that $\sum_{s\in S} w_s \leqslant w_i$.

Given any valid primal solution to the extended model, in particular solutions in which the variables corresponding to dual cuts take a positive value, it is always possible to build a solution with the same cost that is expressed as a non-negative linear combination of valid cutting patterns, which is a solution to Gilmore–Gomory model.

From the primal standpoint, the dual cuts for the cutting stock problem mean that an item of a given size w_i can be cut, and used to fulfill the demand of smaller orders, provided the sum of their widths is smaller than or equal to the initial size. This is done at no cost, because no new rolls are used.

Example 7.1. Consider a variation of Example 5.1 with an unrestricted supply of rolls of size 9 only. As demand widths are equal to 4, 3, and 2, respectively, we can use only dual cuts from sets S of cardinality 1: $-\pi_2 + \pi_1 \leqslant 0$, $-\pi_3 + \pi_2 \leqslant 0$, and of cardinality 2: $-\pi_1 + 2\pi_3 \leqslant 0$. The extended model for the cutting stock problem is given in Fig. 7. Again, only maximal patterns, where the amount of loss is smaller than the width of the smallest roll, are presented. Clearly, the λ_k^r variables in this model and the μ_k^r variables used in Fig. 3 are the same variables. There, they were denoted that way, because they were associated with cycles, while here they are seen as paths between node 0 and 9.

Proposition 7.2 [5]. *The dual cuts introduced in Proposition 7.1 are valid inequalities to the space of optimal solutions of the dual of the one-dimensional cutting stock problem.*

This result was proved showing that any optimal solution to Gilmore–Gomory model must obey all the dual cuts, otherwise there would be a valid cutting pattern separating the optimal solution from the valid dual space of the problem, and contradicting the fact that the solution was valid to the original space. The following example illustrates how the dual cuts effectively cut portions of the dual space of the original model and preserve the set of optimal dual solutions of the cutting stock problem.

w_i	λ_1^1	λ_1^2	λ_1^3	λ_1^4	λ_1^5	λ_1^6	λ_1^7	ν_1	ν_2	ν_3		
w_i =4	2	1	1					-1		-1	$\geq$	20
3		1		3	2	1		1	-1		$\geq$	10
2		1	2		1	3	4		1	2	$\geq$	20

Fig. 7. Extended model.

Example 7.2. Let us consider the family of cutting stock problems with rolls of size 10, and items of size 4 and 3, with instances that result from a particular choice of the values of demands for each size. The valid packings are the solutions to a knapsack problem, and are described by the following set:

$$K = \{(a_1, a_2) : 4a_1 + 3a_2 \leqslant 10,\ a_1, a_2 \geqslant 0 \text{ and integer}\}.$$

The maximal valid packings are: $\{(2,0), (1,2), (0,3)\}$, and the corresponding dual space is defined by the following set $D = \{(\pi_1, \pi_2) : 2\pi_1 \leqslant 1, \pi_1 + 2\pi_2 \leqslant 1, 3\pi_2 \leqslant 1\}$, which is represented in Fig. 8. Notice that packings that are not maximal correspond to redundant dual constraints.

Gilmore and Gomory [18] showed that the dual constraints $\pi_1 \geqslant 0$ and $\pi_2 \geqslant 0$ were valid dual constraints, because we can always replace the equality constraints in the primal problem by greater than or equal to inequalities.

The constraint $\pi_1 \geqslant \pi_2$ is also a valid dual constraint that does not eliminate any optimal dual solution. Note that the dual points denoted as A and D in Fig. 8 can never be optimal dual solutions, because the

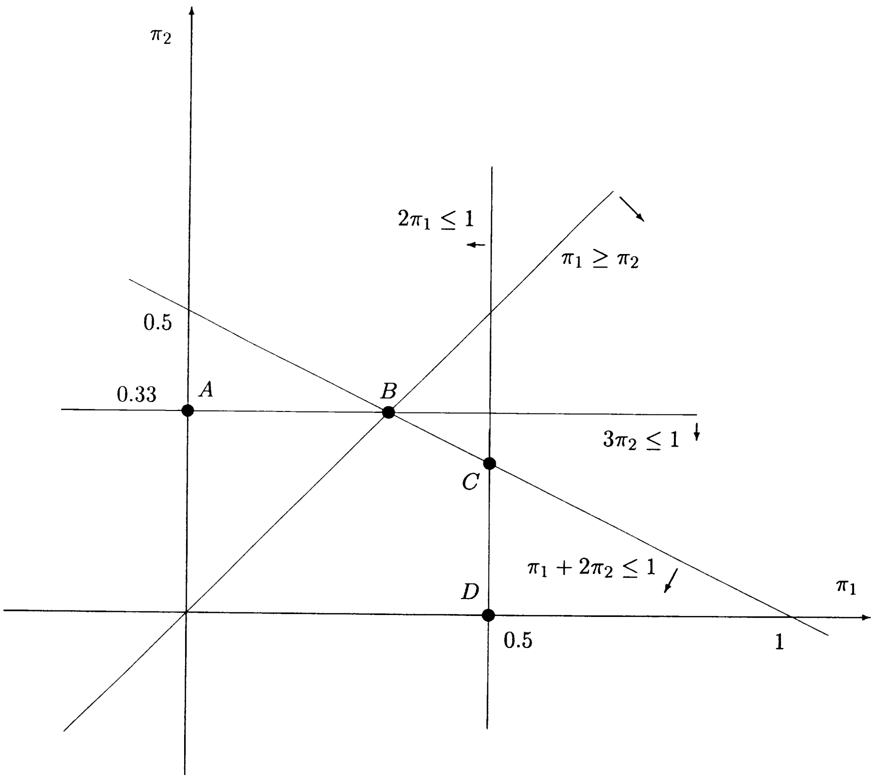

Fig. 8. Dual space of a cutting stock instance.

vector of demands of items, (b_1, b_2), is strictly positive, and the points B and C always dominate A and D, respectively.

The variables in the extended model can also be seen as entities in the graph that defines the arc flow model. A column in Gilmore–Gomory's model is an extreme ray that corresponds to a circulation in the arc flow model. As shown above, there is also a one-to-one correspondence between the circulations and paths between two well defined nodes in the graph. Actually, the variables in classical Gilmore–Gomory model can be seen as paths between nodes 0 and $W_{\max}$ in the acyclic graph. There is also an interesting interpretation to the meaning of the dual cuts when we look at the arc flow model. In the graph, the dual cuts correspond to cycles in the space of the arc flow variables, in which exactly one arc is traversed in the direction opposite to its orientation.

Example 7.3. Consider the graph presented in Fig. 1, and the two following cycles: the first is defined by the arcs $(0, 2)$, $(2, 4)$ and arc $(0, 4)$ traversed in the direction opposite to its orientation, and the second by the arcs $(4, 6)$, $(6, 8)$ and arc $(4, 8)$ traversed in the direction opposite to its orientation. The application of the same Dantzig–Wolfe decomposition presented in Proposition 5.1 would yield for both cycles the variable v_3 in Fig. 7.

In the arc flow model, those cycles are null cost cycles. Combining a path and a cycle (clearly, the path must contain the arc in the cycle traversed in the direction opposite to its orientation) produces a new valid path, and that does not relax the original model.

Other cycles in that graph in which more than one arc is traversed in the direction opposite to its orientation do not produce valid cuts, in the sense defined above, leading to relaxations of the model, that will provide worse LP lower bounds.

That gives a very nice insight into the extended model with Gilmore–Gomory variables and dual cuts. The model has a set of constraints that enforce the satisfaction of demand, and column variables that express the quantities produced for each order. The Gilmore–Gomory variables use full rolls to produce a set of items, while dual cuts use items that are further split to produce shorter items.

That insight provides a nice relation between the arc flow model and onecut models. When an item of an ordered width is split to produce exactly two items of two ordered widths, we have a onecut variable. From this point of view, there are onecut variables that correspond to cycles in the arc flow model in which one arc is traversed in the direction opposite to its orientation. However, the onecut variables that involve residual widths that are not demand widths cannot be inserted in the model.

8. Bin packing as a special case of a vehicle routing problem

In the arc flow model, the flow in an arc contributes to the demand of a given client. In the following model, the demand is satisfied when there is a flow into a node.

Desaulniers et al. [9] observe that the bin packing problem can be viewed as special case of the vehicle routing problem. In particular, in what follows, the bin packing problem will be stated as a special case of the single depot vehicle scheduling problem, when the capacities of all vehicles are identical and the time window constraints are relaxed [11].

The problem is defined in a graph where the items correspond to clients and the bins to vehicles. Clients are located in vertices of a graph, and the bins are the vehicles that visit the vertices, traversing the arcs that join clients, in a route that starts and ends at the depot. Each client demands a load with a value that corresponds to the size of the item, while the capacity of bins is represented by the capacity of the vehicles.

The problem can be represented as an acyclic directed graph in the plane if the depot node is duplicated into two nodes o and d, representing, respectively, the origin and destination of all feasible routes. A route corresponds to a feasible packing of items destined to the clients that are visited by the vehicle. Let N be the set of nodes, each node representing an item, and let $V = N \cup \{o\} \cup \{d\}$.

In the bin packing problem, as we are not interested in packings where an item is present more than once, the directed arcs (i,j) that join an item i to items j of smaller width are admissible, but loops (a directed arc with head and tail at the same node) are not allowed. The set of admissible arcs, denoted as A, comprises the aforementioned arcs, the arcs $(o \times N)$, and the arcs $(N \times d)$.

The objective is to minimize the number of vehicles used, so all trip distances between nodes, c_{ij}, are equal to 0. Let K be the number of vehicles, indexed by k, and assume that they are assuredly sufficient to visit all the clients, i.e., K is an upper bound on the value of the optimum solution. Let x_{ij}^k be a binary variable that denotes the flow of vehicle k in arc $(i,j) \in A : x_{ij}^k = 1$ means that vehicle k visits client j after visiting client i, and $x_{ij}^k = 0$, otherwise.

We consider the case where all bins have equal capacity W. Let w_i denote the width of item $i\ \forall i \in N$. We can also consider that the origin and destination nodes require a null width, i.e., $w_o = w_d = 0$. To model the capacity constraint of each vehicle k, it is necessary to consider additional variables W_i^k, $k = 1,\dots,K\ \forall i \in N$, that denote the portion of the bin k that has been used to pack items up to item i.

The capacity constraints can be modeled using the following non-linear constraints:

$$x_{ij}^k(W_i^k + w_j - W_j^k) \leqslant 0 \quad \forall (i,j) \in A^k,\ \ k = 1,\dots,K, \tag{49}$$

$$w_i \leqslant W_i^k \leqslant W^k \quad \forall i \in V^k,\ \ k = 1,\dots,K. \tag{50}$$

The constraints only allow the visit to client j, if the load accumulated up to client i still leaves enough space to pack the load demanded by client j. The non-linear constraints do not constitute a problem, because, in the reformulated model that results from applying a Dantzig–Wolfe decomposition, they are enforced in the subproblem, and are not present in the master problem, and, again, the model is mainly interesting because it provides a finite branching scheme to be used in a branch-and-price framework.

Example 8.1. Consider an instance of the bin packing problem with 4 items of sizes 4, 3, 2 and 2, respectively. The underlying graph that depicts this instance is presented in Fig. 9. Each vertex represents a different item, and a valid packing corresponds to a directed path between vertices o and d. It is only

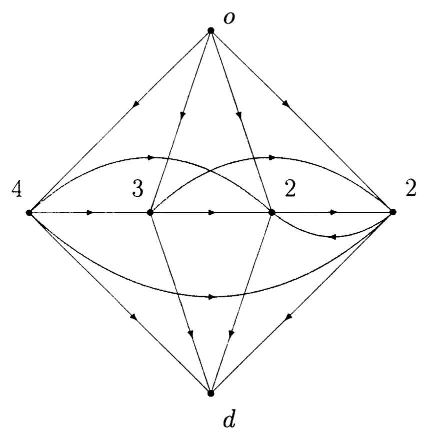

Fig. 9. Graph for an instance of a bin packing.

necessary to consider arcs (i,j) between items such that $w_i \geqslant w_j$, if we search valid packings in which the items are ordered in decreasing values of size. Clearly, the constraints that correspond to the capacity of the items are not represented in the figure.

We state the bin packing problem as the problem of minimizing the number of vehicles leaving the origin node o that are necessary to fulfill the demand of all clients:

$$\min \quad \sum_{k \in K} \sum_{(o,j) \in A^k} x_{oj}^k \tag{51}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j:(i,j) \in A^k} x_{ij}^k = 1 \quad \forall i \in N, \tag{52}$$

$$\sum_{(i,j) \in A^k} x_{ij}^k - \sum_{(i,j) \in A^k} x_{ji}^k = 0 \quad \forall i \in N^k, \ k = 1,\ldots,K, \tag{53}$$

$$x_{ij}^k (W_i^k + w_j - W_j^k) \leqslant 0 \quad \forall (i,j) \in A^k, \ k = 1,\ldots,K, \tag{54}$$

$$w_i \leqslant W_i^k \leqslant W \quad \forall i \in V^k, \ k = 1,\ldots,K, \tag{55}$$

$$x_{ij}^k \text{ binary} \quad \forall (i,j) \in A^k, \ k = 1,\ldots,K. \tag{56}$$

Constraints (52) enforce that each item is visited by exactly one bin. As explained above, constraints (54) and (55) guarantee that the capacity of the bins is obeyed.

A Dantzig–Wolfe decomposition can be applied to this model, keeping constraints (52) in the master problem, while the remaining constraints define the subproblem, which is a longest path problem in an acyclic graph.

A branching scheme based on the arc variables, x_{ij}, was successfully used to solve the vehicle routing problem with time windows [10]. A similar scheme can also be used to find integer solutions to the bin packing problem. To our knowledge, such an approach has never been attempted.

9. Conclusions

In integer programming and combinatorial optimization, many arguments are often derived from the insight given by the different ways of defining the variables and by the structure of the models.

In this paper, we review several LP models that can be used to address the integer solution of one-dimensional cutting stock and bin packing problems. Gilmore–Gomory model does not have the symmetry problems arising in other models and has a polynomial number of constraints, making it more suitable as a tool in a branch-and-price framework.

That does not mean that the other models are useless. Apart from the interest in modeling specific characteristics of some bin packing and cutting stock problems that might arise in real world environments, the other models can be interesting for their use in deriving branching schemes for branch-and-price algorithms that preserve the structure of the subproblem, providing finite algorithms with guaranteed convergence.

Acknowledgements

The author is grateful to Professors Waescher and Stadtler for the stimulating discussion, and for providing references [13,15] which helped enlightening the relations between the models described in this paper. We thank two anonymous referees for their constructive comments, which led to a clearer presentation of the material.

This work was partially supported by Fundação para a Ciência e a Tecnologia (Projecto POCTI/1999/SRI/35568) and Centro de Investigação Algoritmi da Universidade do Minho, and was developed in the Grupo de Engenharia Industrial e de Sistemas.

References

[1] R. Ahuja, T. Magnanti, J. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[2] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, Branch-and-price: Column generation for solving huge integer programs, Operations Research 46 (1998) 316–329.
[3] J.E. Beasley, An exact two-dimensional non-guillotine cutting tree search procedure, Operations Research 33 (1985) 49–64.
[4] J.M. Valério de Carvalho, Exact solution of bin-packing problems using column generation and branch-and-bound, Annals of Operation Research 86 (1999) 629–659.
[5] J.M. Valério de Carvalho, Using dual cuts to accelerate column generation processes, Working paper, Departamento de Produção e Sistemas, Universidade do Minho, 2000. Available from: http://www.eng.uminho.pt/~dps/vc/).
[6] J.M. Valério de Carvalho, A note on branch-and-price algorithms for the one-dimensional cutting stock problem, Computational Optimization and Applications 21 (3) (2002) 339–340.
[7] J. Pinho de Sousa, Time-indexed formulations of non-preemptive single machine scheduling problems, Ph.D. Thesis, Université Catholique de Louvain, Louvain-la-Neuve, 1989.
[8] J. Pinho de Sousa, L.A. Wolsey, A time-indexed formulation of non-preemptive single machine scheduling problems, Mathematical Programming 54 (1992) 353–367.
[9] G. Desaulniers, J. Desrosiers, I. Ioachim, M. Salomon, F. Soumis, D. Villeneuve, A unified framework for deterministic time constrained routing and crew scheduling problems, in: T.G. Crainic, G. Laporte (Eds.), Fleet Management and Logistics, Kluwer Academic Publishers, Boston, 1998, pp. 57–93.
[10] M. Desrochers, J. Desrosiers, M. Salomon, A new optimization algorithm for the vehicle routing problem with time windows, Operations Research 40 (1992) 342–354.
[11] J. Desrosiers, Y. Dumas, M.M. Salomon, F. Soumis, Time constrained routing and scheduling, in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nenmhauser (Eds.), Handbooks in Operations Research and Management Science, vol. 8, Network Routing, Elsevier, Amsterdam, 1995, pp. 35–139.
[12] H. Dyckhoff, A new linear programming approach to the cutting stock problem, Operations Research 29 (1981) 1092–1104.
[13] H. Dyckhoff, Production theoretic foundations of cutting and related processes, in: G. Fandel, H. Dyckhoff, J. Reese (Eds.), Essays in Production Theory and Planning, Springer, Berlin, 1988, pp. 151–180.
[14] H. Dyckhoff, A typology of cutting and packing problems, European Journal of Operational Research 44 (1990) 145–159.
[15] H. Dyckhoff, Bridges between two principal model formulations for cutting stock processes, in: G. Fandel, H. Gehring (Eds.), Beiträge zur Quantitativen Wirtschaftsforschung, Springer, Berlin, 1991, pp. 377–385.
[16] A.M. Geoffrion, Lagrangian relaxation and its uses in integer programming, Mathematical Programming Study 2 (1974) 82–114.
[17] P. Gilmore, Cutting stock, linear programming, knapsacking, dynamic programming and integer programming, some interconnections, in: P.L. Hammer, E.L. Johnson, B.H. Korte (Eds.), Annals of Discrete Mathematics, vol. 4, North-Holland, Amsterdam, 1979, pp. 217–236.
[18] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem, Operations Research 9 (1961) 849–859.
[19] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem – part II, Operations Research 11 (1963) 863–888.
[20] E.L. Johnson, Modeling and strong linear programs for mixed integer programming, in: M. Akgül, H. Hamacher, S. Tüfekçi (Eds.), NATO ASI Series F, vol. 51, Springer, Berlin, 1989, pp. 1–43.
[21] L. Kantorovich, Mathematical methods of organising and planning production (translated from a report in Russian, dated 1939), Management Science 6 (1960) 366–422.
[22] L. Lasdon, Optimization Theory for Large Systems, Macmillan Series in Operations Research, London, 1970.
[23] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York, 1990.
[24] G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, Wiley, New York, 1988.
[25] H. Stadtler, A comparison of two optimization procedures for 1- and $1\frac{1}{2}$-dimensional cutting stock problems, OR Spektrum 10 (1988) 97–111.

[26] H. Stadtler, One-dimensional cutting stock problem in the aluminium industry and its solution, European Journal of Operational Research 44 (1990) 209–224.
[27] P. Scholl, R. Klein, C. Juergens, Bison: A fast hybrid procedure for exactly solving the one-dimensional bin-packing problem, Computers and Operations Research 24 (1997) 627–645.
[28] P. Schwerin, G. Waescher, A new lower bound for the bin-packing problem and its integration into MTP, Pesquisa Operacional 19 (1999) 111–129.
[29] P. Vance, Branch-and-price algorithms for the one-dimensional cutting stock problem, Computational Optimization and Applications 9 (1998) 211–228.
[30] P. Vance, C. Barnhart, E.L. Johnson, G.L. Nemhauser, Solving binary cutting stock problems by column generation and branch-and-bound, Computational Optimization and Applications 3 (1994) 111–130.
[31] F. Vanderbeck, Computational study of a column generation algorithm for binpacking and cutting stock problems, Mathematical Programming A 86 (1999) 565–594.
[32] F. Vanderbeck, On Dantzig–Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm, Operations Research 48 (2000) 111–128.
[33] F. Vanderbeck, L. Wolsey, An exact algorithm for IP column generation, Operations Research Letters 19 (1996) 151–159.
[34] G. Waescher, T. Gau, Heuristics for the integer one-dimensional cutting stock problem: A computational study, OR Spektrum 18 (3) (1996) 131–144.