

An Integer Cutting-Plane Procedure for the Dantzig-Wolfe Decomposition: Theory

by

Troels Martin Range

Discussion Papers on Business and Economics
No. 10/2006

FURTHER INFORMATION
Department of Business and Economics
Faculty of Social Sciences
University of Southern Denmark
Campusvej 55
DK-5230 Odense M
Denmark

Tel.: +45 6550 3271

Fax: +45 6615 8790

E-mail: lho@sam.sdu.dk

<http://www.sam.sdu.dk/depts/virkl/about.shtml>

ISBN 87-91657-09-1

An Integer Cutting-Plane Procedure for the Dantzig-Wolfe Decomposition: Theory

Troels Martin Range*

July 21, 2006

Abstract

The Dantzig-Wolfe decomposition has been extended to Integer Linear Programming (ILP) and Mixed Integer Linear Programming (MILP). Dynamic Column Generation is usually applied to construct an LP-optimal solution for the LP relaxation of the master problem. Polyhedral cuts or branching are the common methods for recovering from a fractional solution. This paper introduces a new procedure to generate violated valid integer inequalities to the master problem of the integer Dantzig-Wolfe decomposition. This procedure uses an auxiliary LP in which standard Gomory Fundamental Cuts can be applied. The approach is generic in the sense that any valid integer Cutting-Plane can be used, i.e. the approach can be extended to other classes of inequalities, even though we limit our attention to Gomory Fundamental Cuts. Using such inequalities will add to the variety of inequalities used in collaboration with Dynamic Column Generation and will therefore improve the bound obtained by the LP relaxation of the master problem.

1 Introduction

Many optimization problems¹ can be formulated as Integer Linear Programs (ILP) or as Mixed Integer Linear Programs (MILP). This is often convenient because a general-purpose ILP or MILP solver can be used to identify an optimal solution. This can, however, be problematic as solving ILP or MILP in general is \mathcal{NP} -hard. A widely used method is to relax the integrality of the variables and then solve the LP relaxation to obtain a lower bound on the optimal-solution value. The LP-optimal solution may have fractional components and may therefore be infeasible for the ILP or MILP. Two basic approaches are used to recover from the infeasible solution. The first is to identify violated valid inequalities and add these to the LP relaxation. This is usually referred to as a Cutting-Plane algorithm. The second approach is to embed the LP relaxation in a Branch-and-Bound algorithm and then solve a series of LP problems with a smaller set of LP-feasible solutions. The combination of these two approaches is usually referred to as Branch-and-Cut. See Jünger et al. (1995) for a detailed description of Branch-and-Cut and Cornuéjols (2006) for a detailed description of valid inequalities for MILP. The reader is also referred to Nemhauser & Wolsey (1988) who give an elaborate treatment of integer programming. Another approach is to utilize

*Department of Business and Economics, Faculty of Social Sciences, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark. E-mail: tra@sam.sdu.dk

¹We will limit our attention to the case of minimization problems, as maximization problems can easily be transformed into minimization problems.

the Dantzig-Wolfe decomposition for integer programs, where the original problem is divided into a master problem and a set of pricing problems. It was introduced by Dantzig & Wolfe (1960) for Linear Programming (LP), i.e. it did not handle integer variables, but Vanderbeck (1995) and Barnhart et al. (1998) extended the theory of the Dantzig-Wolfe decomposition to ILP. The integrality of the master problem is relaxed, whereas the pricing problems are not necessarily relaxed. This yields a tighter lower bound for the original problem than the ordinary LP relaxation. The number of variables in the master problem will be exponential in size and they are therefore generated dynamically. The process of generating variables is usually referred to as Column Generation and is described in detail by Desaulniers et al. (2005). The solution of the LP-relaxed master problem may yield a fractional solution which is infeasible for the original problem. In order to identify a feasible solution it is necessary to recover from this infeasible solution. As for the ordinary LP relaxation we can do this by adding further valid inequalities or by performing branching. Both of these methods are, however, more complex than in the case of Branch-and-Cut. The issue of branching is treated by Vanderbeck (2000) and Vanderbeck (2005a), and we will not investigate the subject further in this paper. When the integer Dantzig-Wolfe decomposition is embedded in a Branch-and-Bound, it is referred to as Branch-and-Price. Using cuts in the master problem is difficult, because the coefficients of each cut have to be available for the pricing problems in order to generate columns. This difficulty usually prohibits the use of integer cutting planes generated directly from the master problem. Instead, it has become common practice to use problem-specific cutting planes, which can be stated with coefficients of the original ILP or MILP. Ralphs & Galati (2006) give an interesting description of different approaches to generate cuts dynamically for Column Generation. The combination of a Cutting-Plane algorithm and Column Generation is referred to as Price-and-Cut and when embedding a Price-and-Cut into a Branch-and-Bound it is referred to as Branch-Price-and-Cut.

The purpose of this paper is to describe a general integer Cutting-Plane procedure, which can be applied to the integer Dantzig-Wolfe decomposition. This procedure will use the Gomory Fundamental Cuts, but is not limited to these cuts. A general integer Cutting-Plane algorithm for the Dantzig-Wolfe decomposition is of interest for two main reasons. First, by developing a general integer Cutting-Plane framework we have a way of tightening the lower bound of the LP relaxation of the Dantzig-Wolfe master problem. It can be applied to a wide variety of problems as we only assume that the variables are integer and we do not assume special structures of the constraints. The use of cutting planes is usually restricted to problem-specific inequalities derived from specific information about the problem. These are not always sufficient to tighten the LP relaxation and therefore branching is usually applied to obtain an integer solution. If we have an integer Cutting-Plane algorithm, we may tighten the LP relaxation further and we limit the use of branching. Second, it is often a good idea to apply different classes of inequalities as they will tend to have a combined effect on the LP relaxation. Valid inequalities derived only by the fact that the solution has to be integer will add to the variety of these classes. We are currently initializing a computational study and the results from this study will be the subject of a future paper.

The rest of the paper is organized as follows. In section 2 we will describe the Dantzig-Wolfe decomposition further as well as the problem of constructing integer cutting planes in this decomposition. Then, in section 3, we introduce the new approach for generating integer cutting planes for the integer Dantzig-Wolfe decomposition. Next, in section 4, we extend the integer Cutting-Plane approach to three specific structures of the Dantzig-Wolfe decomposition. We finish the paper with directions for future research (section 5) and concluding

remarks (section 6).

2 Notation and Theory

Let $\mathbf{x}^t = [x_1, \dots, x_n]$ be a vector of variables and $\mathcal{I} = \{1, \dots, n\}$ be the index set of variables. We let $\mathbf{c} = [c_1, \dots, c_n]$ be an integer cost vector. Let \mathbf{A}_i be an $m_i \times n$ matrix with integer elements for $i = 0, 1$ and let \mathbf{b}_i be an m_i vector integer vector for each $i = 0, 1$. Then we can define the Integer Linear Program (ILP)

$$\begin{aligned} \min \quad & \mathbf{c}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}_0\mathbf{x} = \mathbf{b}_0 \\ & \mathbf{A}_1\mathbf{x} = \mathbf{b}_1 \\ & \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x} \in \mathbb{Z}^n \end{aligned} \tag{1}$$

We will refer to (1) as the original problem. To ease the notation we define

$$X_i = \{\mathbf{x} \in \mathbb{Q}^n \mid \mathbf{A}_i\mathbf{x} = \mathbf{b}_i, \mathbf{x} \geq \mathbf{0}\}, \quad i = 0, 1 \tag{2}$$

which is a rational polytope. As described in the previous section we can solve the original problem by relaxing it and then tighten the relaxation either by generating valid inequalities or by performing branching. In this section we first describe the Cutting-Plane approach, then we describe the Dantzig-Wolfe decomposition for integer programs and, finally, we discuss the difficulty of combining the Cutting-Plane approach with the Dantzig-Wolfe decomposition.

The Cutting-Plane algorithm relaxes the integrality of (1) and then tries to estimate the integer hull by adding further valid inequalities. An inequality $\mathbf{f}\mathbf{x} \geq \mathbf{f}_0$ is valid if $X_0 \cap X_1 \cap \mathbb{Z}^n \subseteq \{\mathbf{x} \in \mathbb{Q}^n \mid \mathbf{f}\mathbf{x} \geq \mathbf{f}_0\}$. Given a solution $\bar{\mathbf{x}}$ to the LP relaxation of (1), we say that $\mathbf{f}\mathbf{x} \geq \mathbf{f}_0$ is violated by $\bar{\mathbf{x}}$ if $\mathbf{f}\bar{\mathbf{x}} < \mathbf{f}_0$. A classical example of valid inequalities for integer programs is given by Gomory (1963a) and Gomory (1963b) who provide an algorithm for identifying such violated integer inequalities for ILP. The Gomory Fundamental Cut is further described by Garfinkel & Nemhauser (1972) and is derived from the simplex tableau. Let \mathcal{B} be the index set of the basic variables and let \mathcal{N} be the index set of the non-basic variables. Let \mathbf{B} and \mathbf{N} be the matrices containing the coefficients of the basic variables and non-basic variables, respectively. The values of the basic variables are calculated as $x_{\mathcal{B}} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}x_{\mathcal{N}}$. The i 'th row of the column vector $\mathbf{B}^{-1}\mathbf{b}$ is denoted α_{i0} , whereas the element in the i 'th row and j 'th column is denoted α_{ij} . Then the fundamental cut is defined as

$$\sum_{j \in \mathcal{N}} ([h]\alpha_{ij} - [h\alpha_{ij}])x_j \geq [h]\alpha_{i0} - [h\alpha_{i0}] \tag{3}$$

where \mathcal{N} is the index set of non-basic variables and h is some non-zero scalar. A typical selection of h is to put $h = 1$, but other values may yield stronger inequalities (see Garfinkel & Nemhauser (1972)). A cut can be identified by investigating the right-hand side of (3), i.e. if $[h]\alpha_{i0} - [h\alpha_{i0}] > 0$, then at least one of the non-basic variables has to be increased from zero. Note that when $h = 1$ and the i 'th element of $x_{\mathcal{B}}$ is fractional, then $\alpha_{i0} - [\alpha_{i0}] > 0$. Hence, for each fractional basic variable a violated fundamental cut exists. In this paper, we are interested in valid inequalities generated from the simplex tableau, because they provide local information about the solution $\bar{\mathbf{x}}$, and different classes of inequalities have been derived

from this information. Cornuéjols (2006) gives a review of state-of-the-art (mixed) integer valid inequalities.

The Cutting-Plane algorithm can tighten the lower bound of the LP relaxation. Another approach to tighten the lower bound is the Dantzig-Wolfe decomposition for integer programs. It can be derived from the following observation. Nemhauser & Wolsey (1988) argue that we can write any point from $\mathbf{x} \in \mathbf{X}_1 \cap \mathbb{Z}^n$ as

$$\begin{aligned} \mathbf{x} &= \sum_{p \in \mathcal{P}} \lambda_p \mathbf{x}_p + \sum_{r \in \mathcal{R}} \delta_r \mathbf{y}_r, \\ \sum_{p \in \mathcal{P}} \lambda_p &= 1, \\ \lambda_p &\geq 0, & \lambda_p &\in \mathbb{Z}, \quad p \in \mathcal{P}, \\ \delta_r &\geq 0, & \delta_r &\in \mathbb{Z}, \quad r \in \mathcal{R} \end{aligned} \tag{4}$$

where \mathcal{P} is the finite index set of integer points within $\text{conv}\{X_1 \cap \mathbb{Z}^n\}$, and \mathcal{R} is the finite index set of integer rays within $\text{cone}\{X_1 \cap \mathbb{Z}^n\}$. It is convenient to assume that $p < r$ for all $p \in \mathcal{P}$ and all $r \in \mathcal{R}$. Each of the points in $\text{conv}\{X_1 \cap \mathbb{Z}^n\}$ is denoted \mathbf{x}_p , and each of the rays of $\text{cone}\{X_1 \cap \mathbb{Z}^n\}$ is denoted \mathbf{y}_r . Note that $\lambda_p \in \{0, 1\}$ because of the convexity constraint. Vanderbeck (1995) uses (4) to obtain the following reformulation of the original problem

$$\begin{aligned} z(\mathcal{P}, \mathcal{R}) &= \min \sum_{p \in \mathcal{P}} c_p \lambda_p + \sum_{r \in \mathcal{R}} c_r \delta_r \\ \text{s.t.} \quad &\sum_{p \in \mathcal{P}} \mathbf{a}_p \lambda_p + \sum_{r \in \mathcal{R}} \mathbf{a}_r \delta_r = \mathbf{b}_0 \\ &\sum_{p \in \mathcal{P}} \lambda_p = 1 \\ &\lambda_p \in \{0, 1\}, & p &\in \mathcal{P} \\ &\delta_r \in \mathbb{Z}_+, & r &\in \mathcal{R} \end{aligned} \tag{5}$$

where $c_p = \mathbf{c} \mathbf{x}_p$, $c_r = \mathbf{c} \mathbf{y}_r$, $\mathbf{a}_p = \mathbf{A}_0 \mathbf{x}_p$ and $\mathbf{a}_r = \mathbf{A}_0 \mathbf{y}_r$. (5) corresponds to the Dantzig-Wolfe master problem except that the variables are required to be integer. Using (4) to obtain the Dantzig-Wolfe master problems is often referred to as discretization.

The formulation (5) has two main difficulties. First, the sizes of \mathcal{P} and \mathcal{R} may be (and usually are) exponential compared to the number of constraints m_1 or the number of original variables n . Hence, solving even the LP relaxation of (5) may be impossible due to the number of variables. The common practice is, therefore, to apply dynamic Column Generation, where most of the variables are removed from the problem and then reinserted if they can give a potential improvement in the objective value. We denote $\bar{\mathcal{P}} \subseteq \mathcal{P}$ and $\bar{\mathcal{R}} \subseteq \mathcal{R}$ the restricted index sets of points and rays. If we replace the index sets \mathcal{P} and \mathcal{R} with their restrictions $\bar{\mathcal{P}}$ and $\bar{\mathcal{R}}$ in (5), we obtain the so-called Restricted Master Problem (RMP). (5) can then be solved by modifying the sets $\bar{\mathcal{P}}$ and $\bar{\mathcal{R}}$ and proving that no better sets exist.

The second problem is that (5) is an ILP and is therefore considered hard to solve. Instead of solving the ILP directly, the requirement of the variables being integer is relaxed and the LP relaxation is solved instead. Throughout this paper we let z_{lb} be the value of the LP relaxation of (5) by dynamic Column Generation. Let $\boldsymbol{\pi}_0$ be the dual variable vector of the first set of constraints of (5) and π_1 be the dual variable for the convexity constraint of (5). Then dynamic Column Generation identifies negative, reduced cost columns for the

LP-relaxed RMP by solving the pricing problem

$$\left(\begin{array}{ll} \min & (\mathbf{c} - \boldsymbol{\pi}_0 \mathbf{A}_0) \mathbf{x} \\ \text{s.t} & \mathbf{x} \in X_1 \cap \mathbb{Z}^n \end{array} \right) < \pi_1 \quad (6)$$

If the left-hand side is strictly less than π_1 , then a negative, reduced cost column has been identified and can be inserted into the LP relaxation. The problem is that the LP relaxation of (5) may yield a fractional solution, which needs to be integerized. This integerization can be obtained by performing branching or by applying further valid inequalities. Branching can be performed in the master problem by applying a set of partitioning constraints in which the current fractional solution is excluded. These partitioning constraints are usually constructed in terms of the original \mathbf{x} -variables, such that it is easy to calculate the coefficients of the variables in extremal form (5). Range (2006) briefly discusses problems of using branching in Branch-and-Price and Vanderbeck (2000) provides several types of branching in a general Branch-and-Price. We will not elaborate further on the topic of branching in this paper as we are mainly interested in a general Price-and-Cut procedure. Instead we will turn to the generation of valid inequalities. If we did not utilize dynamic Column Generation, it would be possible to utilize standard integer cuts such as Gomory Fractional Cuts or Rounding in the master problem. The dynamic Column Generation makes this difficult, however. Suppose that we can find a violated valid inequality in terms of the restricted variable sets λ_p with $p \in \overline{\mathcal{P}}$ and δ_r with $r \in \overline{\mathcal{R}}$

$$\sum_{p \in \overline{\mathcal{P}}} d_p \lambda_p + \sum_{r \in \overline{\mathcal{R}}} d_r \delta_r \geq e_t \quad (7)$$

and we insert it into the restricted master problem. As the inequality was violated it may have a dual price, $\pi \neq 0$, different from zero. Hence, π has to be included into the pricing problem such that it calculates a correct reduced cost coefficient for the columns generated. This involves two problems. First, we have to be able to calculate the coefficients d_p for $p \in \mathcal{P} \setminus \overline{\mathcal{P}}$ and d_r for $r \in \mathcal{R} \setminus \overline{\mathcal{R}}$. Such a coefficient is clearly dependent on the column generated, and we will just assume that we have some oracle $d(\mathbf{x}) : \mathbb{Q}^n \rightarrow \mathbb{Q}^n$ which can be used to calculate the coefficient correctly. Hence, whenever we have a solution \mathbf{x} we have to call the oracle $d(\mathbf{x})$ to calculate the correct coefficient and this can usually not be handled by exact integer optimization algorithms. The second problem we are faced with is that we have to include this function of \mathbf{x} into the objective of the pricing problem as follows

$$\left(\begin{array}{ll} \min & (\mathbf{c} - \boldsymbol{\pi}_0 \mathbf{A}_0) \mathbf{x} - \pi d(\mathbf{x}) \mathbf{x} \\ \text{s.t} & \mathbf{x} \in X_1 \cap \mathbb{Z}^n \end{array} \right) < \pi_1 \quad (8)$$

We observe that $d(\mathbf{x}) \mathbf{x}$ is a non-linear term (unless $d(\mathbf{x})$ is constant) and the pricing problem will therefore be difficult to solve. When $d(\mathbf{x})$ is not constant, the approach is called Non-Robust Branch-Price-and-Cut. It seems reasonable that using the solution information from the LP-basis of the master problem to construct valid inequalities directly will yield other types of inequalities than using hand-crafted valid inequalities. The problem is that cuts generated from the information of the LP-basis usually require information about the coefficient of *all* non-basic columns too. This information is not directly available when using Column Generation, because of the exponential size of $\mathcal{P} \cup \mathcal{R}$. Hence, the term $d(\mathbf{x})$ becomes non-constant, when utilizing classical integer cuts directly in the master problem. Non-Robust Branch-Price-and-Cut has only received little attention, because it usually involves altering the pricing problem to a more difficult problem, but Nemhauser & Park (1991) and Jepsen

et al. (2006) have used this approach, where they modify the pricing problems such that they can handle certain problem-specific valid inequalities.

We will turn to robust Branch-Price-and-Cut, where $d(\mathbf{x})$ is constant because it has no requirement of altering the pricing problem. Hence, if we can construct a robust integer Cutting-Plane method which can eliminate fractional solution in the master problem, then it is straightforward to extend the Branch-Price-and-Cut with this method.

3 Generating Integer Cuts

Suppose that we have found a lower-bound solution for (5) with value z_{lb} , i.e. we have solved the LP relaxation of (5) and obtained $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\delta}})$, where $\bar{\boldsymbol{\lambda}} = (\bar{\lambda}_1, \dots, \bar{\lambda}_{|P|})$ and $\bar{\boldsymbol{\delta}} = (\bar{\delta}_1, \dots, \bar{\delta}_{|R|})$. If this lower-bound solution is integer, then the solution is feasible and we therefore have an optimal solution of (5). On the other hand, if the solution is fractional, it is infeasible for (5). We can transform the solution $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\delta}})$ into a solution $\bar{\mathbf{x}}$ for the original problem by the transformation (4), i.e. we can write

$$\bar{\mathbf{x}} = \sum_{p \in P} \bar{\lambda}_p \mathbf{x}_p + \sum_{r \in R} \bar{\delta}_r \mathbf{y}_r$$

and we have that $\mathbf{c}\bar{\mathbf{x}} = z_{lb}$. Note that if \mathbf{c} is integer, then any integer feasible solution will have to have an integer objective value. Hence, if z_{lb} is fractional, we can reset $z_{lb} := \lceil z_{lb} \rceil$ to the smallest integer which is at least as large as z_{lb} . In the following we will assume that z_{lb} has an integer value.

Now suppose that $\bar{\mathbf{x}}$ has at least one fractional component. We are interested in a cut which separates $\bar{\mathbf{x}}$ from the set of optimal integer solutions. Hence, we seek an inequality that removes $\bar{\mathbf{x}}$ from the set of LP-feasible solutions which may cut away integer points, but never cuts away the optimal integer point. Hence, we are interested in constructing an automated integer Cutting-Plane algorithm. The Gomory Fractional Cutting-Plane is such a procedure. The problem is that we have to have an LP basis (and non-basis) such that we can write $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_B, \bar{\mathbf{x}}_N]$ and $\bar{\mathbf{x}}$ is a vertex of the polytope $X_0 \cap X_1$. But we know that $\bar{\mathbf{x}}$ is a vertex of the polytope $(\text{conv}\{X_1 \cap \mathbb{Z}^n\} + \text{cone}\{X_1 \cap \mathbb{Z}^n\}) \cap X_0$. Hence, $\bar{\mathbf{x}}$ may be an interior point of a face of $X_0 \cap X_1$, and therefore it may not be possible to identify a usable LP basis. The following example illustrates this problem.

Example: Suppose that we have an two structural integer variables x_1 and x_2 , and we have to solve the following optimization problem

$$\begin{aligned} \min \quad & -2x_1 + x_2 \\ \text{s.t} \quad & -5x_1 + 8x_2 \geq 0 \\ & -5x_1 + 3x_2 \leq 0 \\ & 10x_1 + x_2 \geq 10 \\ & -x_1 + 8x_2 \geq 4 \\ & 10x_1 + 8x_2 \leq 45 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{Z} \end{aligned} \tag{9}$$

which has optimal solution $(x_1^*, x_2^*) = (2, 2)$ with value $z^* = -2$. If we solve the LP relaxation of (9), then we obtain the LP-optimal solution $(x_1, x_2) = (3, 1\frac{7}{8})$ having value $z_{lb} = -4\frac{1}{8}$.

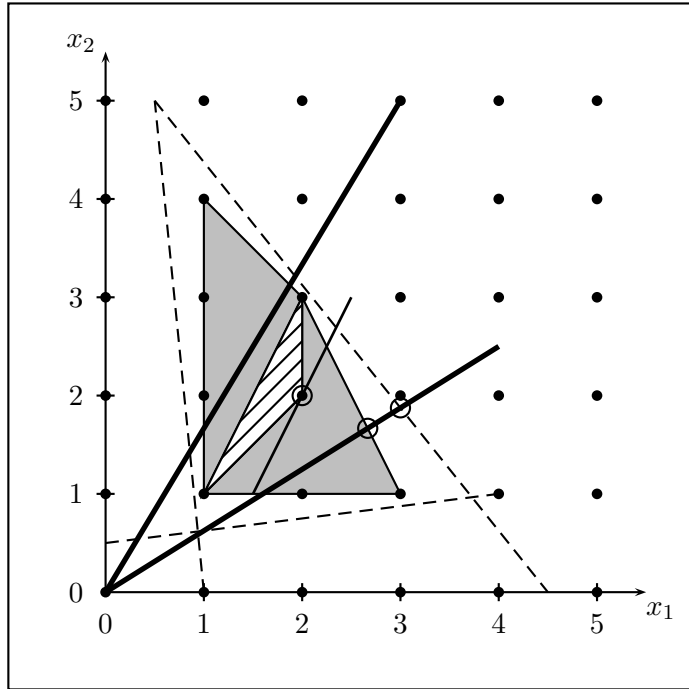


Figure 1: Example of an integer Dantzig-Wolfe decomposition

Instead we use a Dantzig-Wolfe decomposition with the first two constraints in the master problem and the three last constraints in the pricing problem. This is illustrated in figure 1. The bold lines correspond to the boundary of the constraints from the master problem, whereas the dashed lines are the constraints from the pricing problem. The gray area corresponds to the convex hull of the feasible integer points of the pricing problem, and the hatched area is the convex hull of the feasible integer solutions of the full problem. It is interesting to observe that two of the four extreme points of the convex hull of integer points for the pricing problem are not included in the cone generated by constraints of the master problem. Now suppose that we can identify feasible integer points from the pricing problem using some algorithm and we have obtained $\mathbf{x}_1 = [3, 1]^t$ and $\mathbf{x}_2 = [2, 3]^t$. We obtain the LP-optimal solution of the master problem as $(\lambda_1, \lambda_2) = (\frac{2}{3}, \frac{1}{3})$, which yields the solution $\bar{\mathbf{x}} = (2\frac{2}{3}, 1\frac{2}{3})$ with value $-3\frac{2}{3}$. It can be seen in figure 1 that $\bar{\mathbf{x}}$ is not a vertex of the polytope formed by the five constraints of (9). This can also be seen by observing that four of the slack and surplus variables are lifted from their lower bounds and we therefore have one too many basic variables for a basic solution.

Instead, we try to identify another LP solution which resembles $\bar{\mathbf{x}}$ as much as possible and then cut this solution away. Thus, we are interested in constructing a solution \mathbf{x}^* which has the same objective value as $\bar{\mathbf{x}}$ and has as many components equal to $\bar{\mathbf{x}}$ as possible. Define the set $\mathcal{X}_0 = \{j \in \mathcal{I} | \bar{x}_j = 0\}$ as the index set of all the variables having value zero in the solution $\bar{\mathbf{x}}$. This index set will contain the indices of all the variables in the non-basis of the vertex we seek. Hence, we try to retain these variables at a zero value. We also know that z_{lb} is a lower bound value for the original problem, and we therefore require that $\mathbf{c}\mathbf{x}^* \geq z_{lb}$. Thus to

obtain a solution with these properties we solve the problem

$$\begin{aligned}
z^* = \min \quad & \sum_{j \in \mathcal{X}_0} x_j + s_{lb} \\
\text{s.t.} \quad & \mathbf{c}^t \mathbf{x} - s_{lb} = z_{lb} \\
& \mathbf{A}_0 \mathbf{x} = \mathbf{b}_0 \\
& \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1 \\
& \mathbf{x} \geq \mathbf{0}, \quad s_{lb} \geq 0
\end{aligned} \tag{10}$$

which yields a solution (\mathbf{x}^*, s_{lb}^*) . We say that $x_j = 0$ for $j \in \mathcal{X}_0$ and $s_{lb} = 0$ are invariant if we have to identify the solution $(\bar{\mathbf{x}}, \bar{s}_{lb})$. We are interested in identifying a cut, which makes it impossible to obtain the invariant parts of $(\bar{\mathbf{x}}, \bar{s}_{lb})$. In the following we will show that such a cut will separate the solution $(\bar{\mathbf{x}}, \bar{s}_{lb})$ from the set of feasible solutions. First, we show that a solution to (10) exists.

Proposition 3.1 *A solution (\mathbf{x}^*, s_{lb}^*) with $x_j^* = 0, j \in \mathcal{X}_0, s_{lb}^* = 0$ and $z^* = 0$ to problem (10) exists and is optimal.*

Proof: First note that $z^* \geq 0$ because all variables are non-negative. The solution $\bar{\mathbf{x}}$ is an LP-feasible solution with value z_{lb} to (10) with $x_j = 0$ for $j \in \mathcal{X}_0$ and $\bar{s}_{lb} = 0$. Hence, by putting $(\mathbf{x}^*, s_{lb}^*) = (\bar{\mathbf{x}}, \bar{s}_{lb})$, we obtain a solution value $z^* = 0$. ■

From Proposition 3.1 we have that the two solutions $\bar{\mathbf{x}}$ and \mathbf{x}^* can only be distinguished in the variables x_j with $j \in \mathcal{I} \setminus \mathcal{X}_0$. Therefore, we can say that the two solutions resemble each other. We have just shown that the solution $\bar{\mathbf{x}}$ will yield an optimal solution for (10), but we have not shown that $(\bar{\mathbf{x}}, \bar{s}_{lb})$ is an extreme point. Indeed, it may be that $(\bar{\mathbf{x}}, \bar{s}_{lb})$ is not an extreme point, but rather an interior point of a face of the polyhedron

$$P(\mathcal{X}_0, z_{lb}) = \left\{ \mathbf{x} \in \mathbb{Q}_+^n \left| \begin{array}{ll} \mathbf{c}\mathbf{x} - s_{lb} = z_{lb} \\ \mathbf{A}_i \mathbf{x} = \mathbf{b}_i, & i = 0, 1 \\ x_j = 0, & j \in \mathcal{X}_0 \\ s_{lb} = 0 \end{array} \right. \right\}$$

in which case an alternative solution (\mathbf{x}^*, s_{lb}^*) with some elements $x_j^* \neq \bar{x}_j$ for $j \in \mathcal{I} \setminus \mathcal{X}_0$ will be found.

Example: (Continued from above). Let \mathcal{I} be the index set of the structural variables as well as the slack and surplus variables of (9). Then we have that $|\mathcal{I}| = 7$. Now, using the solution $(x_1, x_2) = (2\frac{2}{3}, 1\frac{2}{3})$ obtained from the master problem we have that only the first constraint of (9) is binding, and all structural variables have values larger than zero i.e. $|\mathcal{X}_0| = 1$. Thus we have $|\mathcal{I} \setminus \mathcal{X}_0| = 6 > 5 = m_1 + m_2$ which indicates that one more variable than needed for a basis is positive. We already have a lower bound of $-3\frac{2}{3}$, and we know that the objective value has to be an integer, i.e. we put $z_{lb} = \lceil -3\frac{2}{3} \rceil = -3$ and add the lower bound constraint $-2x_1 + x_2 - s_{lb} = -3$ with $s_{lb} \geq 0$ to the master problem. Resolving the master problem yields the solution $(\lambda_1, \lambda_2) = (\frac{1}{2}, \frac{1}{2})$, which can be transformed into the new solution $\bar{\mathbf{x}} = (2\frac{1}{2}, 2)$. We continue from this solution. Note that only \bar{s}_{lb} has a value of zero and we have that $\mathcal{X}_0 = \emptyset$. We have, however, added the additional constraint $\mathbf{c}\mathbf{x} - s_{lb} = z_{lb}$. Hence, we still have one too many positive variables to construct a basic solution. Now, solving the problem (10) we obtain the solution $(x_1^*, x_2^*, s_1^*, s_2^*, s_3^*, s_4^*, s_5^*, s_{lb}^*) = (2\frac{2}{11}, 1\frac{4}{11}, 0, 6\frac{9}{11}, 13\frac{2}{11}, 4\frac{8}{11}, 12\frac{3}{11}, 0)$. Hence, we have constructed a basic vector $\mathbf{x}_B = [x_1, x_2, s_2, s_3, s_4, s_5]^t$ and a non-basic vector $\mathbf{x}_N = [s_1, s_{lb}]^t$. This solution is within $P(\emptyset, -3)$, but it is clearly fractional.

Note that $P(\mathcal{X}_0, z_{lb})$ contains $(\bar{\mathbf{x}}, \bar{s}_{lb})$ and is therefore non-empty. We are now interested in either identifying an integer feasible point from $P(\mathcal{X}_0, z_{lb})$ or proving that no feasible integer points exist in $P(\mathcal{X}_0, z_{lb})$. This is, clearly, a separation problem where we try to separate the infeasible solutions in $P(\mathcal{X}_0, z_{lb})$ from the set of feasible solutions. The following proposition states an interesting property of an integer solution from $P(\mathcal{X}_0, z_{lb})$.

Proposition 3.2 *An integer optimal solution (\mathbf{x}^*, s_{lb}^*) to problem (10) will yield an upper bound solution to the original formulation (1). Furthermore, if $s_{lb}^* = 0$, then \mathbf{x}^* is an optimal feasible solution to the original problem (1).*

Proof: Suppose that (\mathbf{x}^*, s_{lb}^*) is an integer optimal solution to problem (10). Then we have that $\mathbf{x}^* \in (\{\mathbf{x} \in \mathbb{R}^n | \mathbf{c}\mathbf{x} \geq z_{lb}\} \cap X_0 \cap X_1 \cap \mathbb{Z}^n) \subseteq X_0 \cap X_1 \cap \mathbb{Z}^n$. Hence, \mathbf{x}^* lies in the set of feasible solution of the original problem (1). Therefore, we have that $\mathbf{c}\mathbf{x}^* = z_{ub}$ is an upper bound for (1). If the surplus variable $s_{lb} = 0$, then we know that $\mathbf{c}\mathbf{x}^* = z_{lb}$. But then we have $z_{ub} = z_{lb}$, which proves the optimality of \mathbf{x}^* for problem (1). ■

Proposition 3.2 gives an alternative optimality criterion for the integer Dantzig-Wolfe decomposition and is in that sense an interesting theoretical observation.

A feasible solution to (10) is not necessarily integer, but we know that only integer solutions are valid for (1). Therefore, we initialize a Cutting-Plane procedure to integerize the solution of (10). Here we have the advantage that we know a solution corresponding to a vertex of the feasible set of solutions of (10), and we can therefore use the Gomory Integer Cutting-Plane procedure to obtain an integer cut. We define the Auxiliary Separation Linear Program (ASLP) as

$$\begin{aligned}
z^*(\mathcal{T}) = \min \quad & \sum_{j \in \mathcal{X}_0} x_j \quad + s_{lb} \\
\text{s.t.} \quad & \mathbf{c}^t \mathbf{x} \quad - s_{lb} = z_{lb} \\
& \mathbf{A}_0 \mathbf{x} \quad = \mathbf{b}_0 \\
& \mathbf{A}_1 \mathbf{x} \quad = \mathbf{b}_1 \\
& \mathbf{d}^t \mathbf{x} \quad + d_{lb}^t s_{lb} \geq d_0^t, \quad t \in \mathcal{T} \\
& \mathbf{x} \geq \mathbf{0}, \quad s_{lb} \geq 0
\end{aligned} \tag{11}$$

where T is the index set of integer cuts generated. Clearly, (11) is equivalent to (10) if $\mathcal{T} = \emptyset$, and we have that $z^*(\mathcal{T}_1) \leq z^*(\mathcal{T}_2)$ for $\mathcal{T}_1 \subseteq \mathcal{T}_2$. If the inequalities with indices in T are integer cutting planes, then no integer solution from (10) will be removed from (11).

Proposition 3.3 *A valid inequality for the problem (10) with integer variables \mathbf{x} and s_{lb} will never remove the optimal integer solution to (1).*

Proof: As z_{lb} is a valid lower bound for the original problem (1), we know that adding the constraint $\mathbf{c}\mathbf{x} \geq z_{lb}$ to (1) will never change the optimal solution value. Hence, estimating the integer hull of $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{c}\mathbf{x} \geq z_{lb}\} \cap X_0 \cap X_1 \cap \mathbb{Z}^n$ will never cut away the optimal solution. This is exactly what (11) does. ■

Now, by adding valid integer cutting planes we try to eliminate the possibility of having all x_j -variables with $j \in \mathcal{X}_0$ at zero and at the same time have $s_{lb} = 0$. Hence, we try to eliminate the possibility of using points in the polyhedron $P(\mathcal{X}_0, z_{lb})$.

Proposition 3.4 *If problem (11) has $z^*(T) > 0$ for some T , then the solution $\bar{\mathbf{x}}$ can never be LP-feasible in the original problem (1), when adding the cuts identified in T .*

Proof: Suppose that $z^*(T) = \sum_{j \in D_0} x_j^* + s_{lb}^* > 0$, where (\mathbf{x}^*, s_{lb}^*) is the optimal solution to (11). Then either $x_j^* > 0$ for some $j \in D_0$ or $s_{lb}^* > 0$ because $\mathbf{x} \geq \mathbf{0}$ and $s_{lb} \geq 0$. As (\mathbf{x}^*, s_{lb}^*) is the closest we can get to the invariant values of $(\bar{\mathbf{x}}, \bar{s}_{lb})$ without being equal to these values, then the solution $(\bar{\mathbf{x}}, \bar{s}_{lb})$ will be infeasible for the original problem. As we know that $z^*(\emptyset) = 0$, then there exists at least one cut with index in T , which separates $(\bar{\mathbf{x}}, \bar{s}_{lb})$ from the set of feasible solutions. We can then add the cuts with indices in T to problem (1) to remove the possibility of using $(\bar{\mathbf{x}}, \bar{s}_{lb})$. ■

Now, what we try to identify is a set T such that $z^*(T) > 0$ and thereby prove that at least one of the valid inequalities with indices in T is violated by the solution $(\bar{\mathbf{x}}, \bar{s}_{lb})$. As such a valid inequality will add at most a slack variable or surplus variable we can rewrite the inequalities in terms of the original \mathbf{x} -variables. The inequality, written in terms of the original variables, can again be transformed into terms of the $(\boldsymbol{\lambda}, \boldsymbol{\delta})$ -variables and then be added to the LP relaxation of (5). Hence, we have a robust integer Cutting-Plane algorithm for the integer Dantzig-Wolfe decomposition.

Example: (Continued from above). Now we use the basic vector $\mathbf{x}_B = [x_1, x_2, s_2, s_3, s_4, s_5]$ and the nonbasic vector $\mathbf{x}_N = [s_1, s_{lb}]$ to obtain the Gomory Fundamental Cut (with $h = 1$) $\frac{1}{11}s_{lb} \geq \frac{3}{11}$, which can be transformed into the constraint $-2x_1 + x_2 \geq -2$ (note that it is parallel with the objective function and we can therefore put $z_{lb} = -2$). Inserting this cut into the ASLP yields the solution $(x_1^*, x_2^*, s_1^*, s_2^*, s_3^*, s_4^*, s_5^*, s_6^*, s_{lb}^*) = (1\frac{5}{11}, \frac{10}{11}, 0, 4\frac{6}{11}, 5\frac{5}{11}, 1\frac{9}{11}, 23\frac{2}{11}, 0, 0)$, where s_6 is the surplus variable of the new constraint and $z^* = 0$ (if we did not reset z_{lb} to -2 then we would have $z^* = 1$, and we would have proved that the current solution for the master problem was infeasible). We generate a new Gomory Fundamental Cut $\frac{1}{11}s_1 + \frac{3}{11}s_{lb} \geq \frac{6}{11}$ which can be transformed into the constraint $-x_1 + x_2 \geq 0$. Inserting this constraint into the ASLP and resolving yields the integer solution $(x_1^*, x_2^*, s_1^*, s_2^*, s_3^*, s_4^*, s_5^*, s_6^*, s_7^*, s_{lb}^*) = (2, 2, 6, 4, 12, 10, 9, 0, 0, 0)$ with solution value $z^* = -2$. As $s_{lb}^* = 0$ we have by proposition 3.2 that the solution is optimal. Note that X_1 is bounded, whereas X_0 is unbounded. It is interesting to observe that the cuts generated are not valid integer cuts ($conv\{X_0\} + cone\{X_0\}$) or $conv\{X_1\}$, but are valid for $conv\{X_0 \cap X_1 \cap \{\mathbf{x} \in \mathbb{R}^n | \mathbf{c}\mathbf{x} \geq z_{lb}\}\}$. This suggests that there is unused information, which can be used to generate integer cuts.

Several observations can be made for the approach that we suggest. We summarize these in the following remarks.

Remark: Note the resemblance to the initialization of the Simplex algorithm Phase 0, where we try to identify a basic feasible solution with all artificial variables at zero. This is exactly what we are doing after adding cuts to (11) and if we cannot identify a solution with all invariant variables equal to zero, then we have proved that a solution which has all of these variables at zero is infeasible.

Remark: Our approach will implicitly force variables at their upper bounds to remain at these upper bounds. This can be realized as follows. Let $0 \leq x_j \leq u_j$ and add the upper bound as an explicit constraint, i.e. $x_j + s_j = u_j$, where $s_j \geq 0$. When $x_j = u_j$, then s_j will be zero and its index is added to \mathcal{X}_0 . This observation is especially important in the case

where the x_j variables are required to be binary, i.e. an upper bound $u_j = 1$ is imposed on the binary variables. Binary variables are used in many models involving networks², where they indicate whether or not a connection is used. A significant portion of the binary variables will be on their upper bound in such models, and we can therefore significantly limit the degree of freedom in the corresponding ASLP. This will lead to faster identification of cuts or even stronger cuts.

Remark: It will be easier to identify violated integer inequalities in problems, where the $\bar{\mathbf{x}}$ is degenerate (i.e. having basic variables equal to zero). This is due to our definition of \mathcal{X}_0 , where all indices of variables equal to zero are included. The more elements of $\bar{\mathbf{x}}$ equal to zero, the more indices are added to \mathcal{X}_0 , and therefore we will have less variables with indices in $\mathcal{I} \setminus \mathcal{X}_0$.

Remark: In proposition 3.4 we argued that whenever $z^*(\mathcal{T}) > 0$, then at least one of the cuts with indices in \mathcal{T} is violated. In that case, we can just add the cut to the master problem. It may, however, be prudent to identify additional valid inequalities. We know that for $\mathcal{T} \subseteq \mathcal{S}$ we have that $z^*(\mathcal{T}) \leq z^*(\mathcal{S})$. As long as we have a fractional solution to (11), we may be able to identify additional valid inequalities from $\mathcal{S} \setminus \mathcal{T}$ which may move the invariant elements of $(\bar{\mathbf{x}}, \bar{s}_{lb})$ further away from the corresponding elements of (\mathbf{x}^*, s_{lb}^*) . This is indicated in the objective of the ASLP by $z^*(\mathcal{T}) < z^*(\mathcal{S})$.

In figure 2 we have illustrated the Price-and-Cut using the integer Cutting-Plane method. This method should be embedded into each node in a Branch-and-Bound tree to obtain a Branch-Price-and-Cut using integer cuts, i.e. when the method terminates then we either have a fractional solution for which branching is needed or we have an integer solution. The gray box corresponds to a standard Column Generation, which usually terminates when no columns can be found. If the solution is fractional, the standard Column Generation would normally terminate. Instead we initialize an integer Cutting-Plane algorithm, which constructs an ASLP (11) to identify a feasible LP basis. If the LP solution to the ASLP is fractional, we try to identify violated integer cuts and resolve the ASLP. On the other hand, if the LP solution is integer, we have from proposition 3.2 that the LP solution is an upper-bound solution. In that case, we update the upper-bound value and transform any violated cut found into a master form such that it can be inserted into the RMP. We have not specified which integer cuts should be generated as a range of different cuts exist which can be utilized. Identifying which cuts should be used will be a subject of a future paper describing the computation. Furthermore, we have not specified how many violated cuts that are sufficient to terminate the integer Cutting-Plane algorithm. Clearly, one can terminate the Cutting-Plane algorithm if $z^*(\mathcal{T}) > 0$, but as noted above, it may be worthwhile to continue the Cutting-Plane algorithm to obtain a larger violation. Finally, we have not included bound checking in the diagram, i.e. if we identify an integer solution in the ASLP and we have $z_{lb} = z_{ub}$, then we can terminate the overall algorithm.

The procedure described above can easily be used in the root node of a Branch-and-Bound, but should be used with caution when used after branching. The main reason for this is that the ASLP uses the lower bound constraint $\mathbf{c}\mathbf{x} \geq z_{lp}$ to generate valid integer inequalities.

²Examples of network problems, where the Dantzig-Wolfe decomposition is used, are the Vehicle Routing Problem with Resource Constraints (see Desrochers et al. (1992)), the Elementary Shortest Path Problem with Resource Constraints (see Range (2006)), and Scheduling Problems (see Desaulniers et al. (1998)).

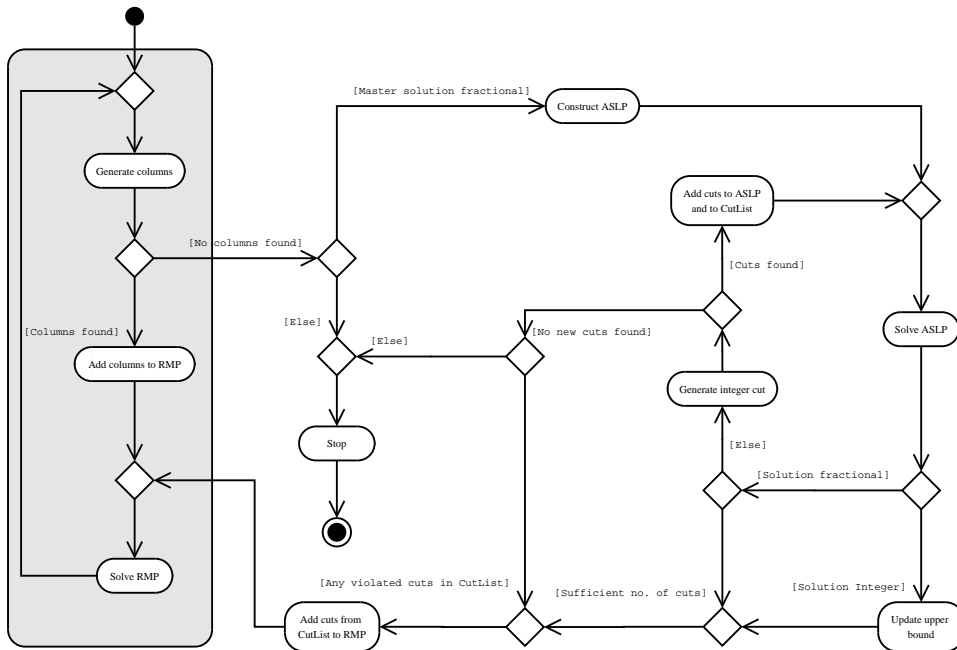


Figure 2: Activity Diagram of the Price-and-Cut Algorithm using integer cuts.

Suppose that we have two nodes in the Branch-and-Bound tree with the lower bounds z_{lp}^1 and z_{lp}^2 and $z_{lp}^1 > z_{lp}^2$. Then it is not valid to use integer inequalities from node 1 in node 2, because the constraint $\mathbf{c}\mathbf{x} \geq z_{lp}^1$ is not necessarily a valid inequality of node 2. This can be omitted by using a global lower bound z_{glb} instead of the local lower bounds, but it will make the generation of violated inequalities harder. Furthermore, we cannot use proposition 3.4 to identify violated cuts. Instead, each of the constraints with indices in \mathcal{T} should be checked directly. On the other hand, a cut identified using only the global lower bound and no branching constraints will be valid for any node in the Branch-and-Bound tree. It may therefore be interesting to use our approach only with the globally valid constraints instead of utilizing the additional information obtained by branching.

In this section, we have described a procedure which can be used to generate violated valid integer inequalities for the Dantzig-Wolfe reformulation of integer programs, where dynamic column generation is used. As of now, we have not conducted a computational study of the method. We are, however, currently preparing such a study.

4 Extensions

The Cutting-Plane approach that we have suggested above can be extended to several interesting cases. In the following we will briefly describe these extensions. Some of these extensions use relaxations of the original problems in order to obtain valid inequalities. Hence, we are not guaranteed an optimality criterion, nor facet inducing valid inequalities. The aim is to describe an ASLP for each extension such that it is possible to utilize valid integer inequalities for these extensions.

In section 4.1, we will describe the straightforward extension to blockangular structures,

where one or more independent blocks of constraints are present in the original problem. When the blocks in such a structure are identical, then the master problem is transformed in such a way that the pricing problems are identical. This will require a similar relaxation of the corresponding ASLP. We will discuss this extension in section 4.2. In some applications, such as resource-constrained routing, not all variables are included in the master problem and we therefore need to relax the ASLP to obtain valid inequalities using the same subset of variables as the master problem. We describe this in section 4.3. The extensions can, of course, be combined, but we limit our attention to the pure forms of the extensions.

4.1 Blockangular Structures

Many problems have structures, where only some constraints span almost all of the variables and the rest of the constraints can be divided into small blocks, where the constraints only span the same small subset of variables. These structures are often referred to as blockangular or block-diagonal³. The Cutting-Plane approach can be extended to blockangular structures in a straightforward manner. Suppose that we have an index set \mathcal{K} of sets of variables and let \mathbf{x}_k for $k \in \mathcal{K}$ be the vector of variables. Furthermore, we define $\mathcal{I} = \cup_{k \in \mathcal{K}} \mathcal{I}(k)$ to be the index set of all variables, where $\mathcal{I}(k)$ are disjoint sets of indices for the variables in \mathbf{x}_k . We let n_k be the length of the vector \mathbf{x}_k . For each $k \in \mathcal{K}$ we associate a cost vector \mathbf{c}_k and two matrices \mathbf{A}_{0k} and \mathbf{A}_k . Finally, we define \mathbf{b}_0 and \mathbf{b}_k as the right-hand side column vectors. We will use X_k in the same way as defined in (2) to denote the polyhedron defined by the constraint set $\mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k$ and the non-negativity constraints. This yields the following integer blockangular minimization problem

$$\begin{aligned} \min \quad & \sum_{k \in \mathcal{K}} \mathbf{c}_k \mathbf{x}_k \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} \mathbf{A}_{0k} \mathbf{x}_k = \mathbf{b}_0 \\ & \mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k, \quad k \in \mathcal{K} \\ & \mathbf{x}_k \geq 0, \quad k \in \mathcal{K} \\ & \mathbf{x}_k \in \mathbb{Z}^{n_k}, \quad k \in \mathcal{K} \end{aligned} \tag{12}$$

which again can be reformulated into the master problem

$$\begin{aligned} \min \quad & \sum_{k \in \mathcal{K}} \left(\sum_{p \in \mathcal{P}(k)} c_{pk} \lambda_{pk} + \sum_{r \in \mathcal{R}(k)} c_{rk} \delta_{rk} \right) \\ \text{s.t.} \quad & \sum_{k \in \mathcal{K}} \left(\sum_{p \in \mathcal{P}(k)} \mathbf{a}_{pk} \lambda_{pk} + \sum_{r \in \mathcal{R}(k)} \mathbf{a}_{rk} \delta_{rk} \right) = \mathbf{b}_0 \\ & \sum_{p \in \mathcal{P}(k)} \lambda_{pk} = 1, \quad k \in \mathcal{K} \\ & \lambda_{pk} \in \{0, 1\}, \\ & \delta_{rk} \in \mathbb{Z}_+ \end{aligned} \tag{13}$$

³A typical example is the Heterogeneous Vehicle Routing Problem, where a set of customers has to be visited by a heterogeneous fleet of vehicles. Each type of vehicle has its own block of constraints, and all the vehicles are subject to the constraint that each customer has to be visited exactly once by exactly one vehicle (see Range (2006) for an example of Heterogeneous Vehicle Routing Problem).

where $c_{pk} = \mathbf{c}_k \mathbf{x}_{pk}$, $c_{rk} = \mathbf{c}_k \mathbf{y}_{rk}$, $\mathbf{a}_{pk} = \mathbf{A}_{0k} \mathbf{x}_{pk}$ and $\mathbf{a}_{rk} = \mathbf{A}_{0k} \mathbf{y}_{rk}$. The vectors \mathbf{x}_{pk} and \mathbf{y}_{rk} are points and rays of the polyhedron X_k for each $k \in \mathcal{K}$. Now define $\mathcal{X}_0(k) = \{j \in \mathcal{I}(k) | x_j = 0\}$. The ASLP will then be

$$\begin{aligned}
\min \quad & \sum_{k \in \mathcal{K}} \sum_{j \in \mathcal{X}_0(k)} x_j + s_{lb} \\
\text{s.t.} \quad & \sum_{k \in \mathcal{K}} \mathbf{c}_k \mathbf{x}_k - s_{lb} = z_{lb} \\
& \sum_{k \in \mathcal{K}} \mathbf{A}_{0k} \mathbf{x}_k = \mathbf{b}_0 \\
& \mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k, \quad k \in \mathcal{K} \\
& \sum_{k \in \mathcal{K}} \mathbf{d}_k^t \mathbf{x}_k + d_{lb}^t s_{lb} \geq d_0^t, \quad t \in \mathcal{T} \\
& \mathbf{x}_k \geq 0, \quad k \in \mathcal{K} \\
& s_{lb} \geq 0
\end{aligned} \tag{14}$$

It is possible to extend propositions 3.1-3.4 in a straightforward manner, and we will omit restating the results here.

What is interesting about this Cutting-Plane approach for blockangular structures is that the cuts will typically span across several of the variable sets. When the constraints $\mathbf{A}_k \mathbf{x}_k = \mathbf{b}_k$ are different, a cut spanning several variable sets will add additional information about the relation between these sets. The process of handcrafting problem-dependent inequalities is often lengthy and is typically inflexible to changes in the structures of the individual blocks. Hence, from a development point-of-view it will be of interest to have a set of problem independent valid inequalities spanning several blocks, such that the development of problem-dependent inequalities will be limited. On the other hand, one of the reasons for using decomposition is that even the LP relaxation of the original problem contains too large sets of variables and constraints. This difficulty is reflected in the ASLP of blockangular structures, where we have almost the same number of constraints and almost the same number of variables as in the original problem. This may be handled by removing some of the variables and some of the constraints from the ASLP and then generating these when needed, i.e. a simple form of Price-and-Cut where we initially keep all the variables not included in any $D_0(k)$ as well as a small set of the zero variables. We have not investigated this approach, but it may be interesting as it will probably speed up the cut generation process significantly.

4.2 Identical Subproblems

A special case of the blockangular structures is where the data for each of the blocks are identical⁴. That is, for each pair $i, j \in \mathcal{K}$ we have that $n_i = n_j$, $\mathbf{c}_i = \mathbf{c}_j$, $\mathbf{A}_{0i} = \mathbf{A}_{0j}$, $\mathbf{A}_i = \mathbf{A}_j$ and $\mathbf{b}_i = \mathbf{b}_j$. We put $n = n_i$, $\mathbf{c} = \mathbf{c}_i$, $\mathbf{A}_0 = \mathbf{A}_{i0}$, $\mathbf{A} = \mathbf{A}_i$ and $\mathbf{b} = \mathbf{b}_i$. Hence, the problem becomes highly symmetric. To overcome the symmetry, the convexity constraints are typically aggregated, which allows the use of a single common pricing problem. That is, we observe that the pricing problems are identical and therefore have the same set of points \mathcal{P} and rays \mathcal{R} , and we therefore put $\mathcal{P} = \mathcal{P}(k)$ and $\mathcal{R} = \mathcal{R}(k)$ for all $k \in \mathcal{K}$. Furthermore, we

⁴The (Homogeneous) Vehicle Routing Problem with Resource Constraints (see Desrochers et al. (1992)) and the Cutting Stock Problem (see Amor & Carvalho (2005)) are prevalent examples of blockangular structures with identical blocks.

put $\lambda_p = \sum_{k \in \mathcal{K}} \lambda_{pk}$ and $\delta_r = \sum_{k \in \mathcal{K}} \delta_{rk}$. Thus the master problem becomes

$$\begin{aligned}
\min \quad & \sum_{p \in \mathcal{P}} c_p \lambda_p & + \sum_{r \in \mathcal{R}} c_r \delta_r \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}} \mathbf{a}_p \lambda_p & + \sum_{r \in \mathcal{R}} \mathbf{a}_r \delta_r = \mathbf{b}_0 \\
& \sum_{p \in \mathcal{P}} \lambda_p & = |\mathcal{K}| \\
& \lambda_p \in \{0, 1\}, & p \in \mathcal{P} \\
& \delta_r \in \mathbb{Z}_+, & r \in \mathcal{R}
\end{aligned} \tag{15}$$

where the integrality is relaxed to obtain the LP relaxation. Then the pricing problem becomes identical to (6), where π_1 is the dual price of the second constraint in the LP relaxation of (15). When the master problem has been aggregated, it will be difficult to differentiate between columns arising from different pricing problems. We suggest doing the following. If we put $\mathbf{y} = \sum_{k \in \mathcal{K}} \mathbf{x}_k$, then we can aggregate the blocks⁵ by

$$|\mathcal{K}| \mathbf{b} = \sum_{k \in \mathcal{K}} \mathbf{A} \mathbf{x}_k = \mathbf{A} \sum_{k \in \mathcal{K}} \mathbf{x}_k = \mathbf{A} \mathbf{y} \tag{16}$$

Clearly, any solution satisfying the individual blocks $\mathbf{A} \mathbf{x}_k = \mathbf{b}$ will also satisfy constraint (16), i.e. by replacing the individual blocks with a single aggregated block we obtain the following relaxation

$$\begin{aligned}
\min \quad & \mathbf{c} \mathbf{y} \\
\text{s.t.} \quad & \mathbf{A}_0 \mathbf{y} = \mathbf{b}_0 \\
& \mathbf{A} \mathbf{y} = |\mathcal{K}| \mathbf{b} \\
& \mathbf{y} \geq \mathbf{0} \\
& \mathbf{y} \in \mathbb{Z}^n
\end{aligned} \tag{17}$$

which will be a significantly smaller optimization problem than the original optimization problem. What is interesting is that an infeasible solution to (17) will also be infeasible for the original problem. Furthermore, we can transform a solution $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\delta}})$ into the \mathbf{y} -variables by observing that

$$\begin{aligned}
\bar{\mathbf{y}} &= \sum_{k \in \mathcal{K}} \bar{\mathbf{x}}_k \\
&= \sum_{k \in \mathcal{K}} \left(\sum_{p \in \mathcal{P}(k)} \bar{\lambda}_{pk} \mathbf{x}_p + \sum_{r \in \mathcal{R}(k)} \bar{\delta}_{rk} \mathbf{y}_r \right) \\
&= \sum_{p \in \mathcal{P}} \bar{\lambda}_p \mathbf{x}_p + \sum_{r \in \mathcal{R}} \bar{\delta}_r \mathbf{y}_r
\end{aligned} \tag{18}$$

⁵This aggregation is similar to the aggregation which is used in the (Homogeneous) Capacitated Vehicle Routing Problem, when transforming a 3-index formulation into a 2-index formulation.

Hence, if we can identify a valid inequality violated by $\bar{\mathbf{y}}$, then we also have a cut which is violated by $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\delta}})$. Now, we can define the associated SLP for identical substructures as

$$\begin{aligned}
z^*(T) = \min \quad & \sum_{j \in \mathcal{Y}_0} y_j \quad + s_{lb} \\
\text{s.t.} \quad & \mathbf{c}\mathbf{y} \quad - s_{lb} = z_{lb} \\
& \mathbf{A}_0\mathbf{y} = \mathbf{b}_0 \\
& \mathbf{A}\mathbf{y} = |\mathcal{K}|\mathbf{b} \\
& \mathbf{d}^t\mathbf{y} + d_{lb}^t s_{lb} \geq d_0^t, \quad t \in T \\
& \mathbf{y} \geq \mathbf{0}
\end{aligned} \tag{19}$$

where $\mathcal{Y}_0 = \{j | \bar{y}_j = 0\}$. It has to be noted that proposition 3.2 does not hold for (19) as it stems from a relaxation of the full problem. Hence, we cannot use (19) to prove optimality for problem (15).

4.3 Hidden Variables

In some cases it is possible to hide a subset of the variables in the pricing problem⁶. In general, we can write a problem as

$$\begin{aligned}
\min \quad & \mathbf{c}\mathbf{x} \\
\text{s.t.} \quad & \mathbf{A}_0\mathbf{x} = \mathbf{b}_0 \\
& \mathbf{A}_1\mathbf{x} = \mathbf{b}_1 \\
& \mathbf{A}_2\mathbf{x} + \mathbf{B}\mathbf{t} = \mathbf{b}_2 \\
& \mathbf{x} \in \mathbb{Z}_+^n \\
& \mathbf{t} \in T
\end{aligned} \tag{20}$$

where T can be the set of non-negative real vectors, the set of integer vectors, or some mix of these. Hence, keeping only the first constraint in the master problem will hide the \mathbf{t} -variables in the pricing problem, i.e. the pricing problem becomes

$$\left(\begin{array}{l} \min \quad (\mathbf{c} - \pi_0 \mathbf{A}_0) \mathbf{x} \\ \text{s.t.} \quad \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1 \\ \quad \mathbf{A}_2 \mathbf{x} + \mathbf{B} \mathbf{t} = \mathbf{b}_2 \\ \quad \mathbf{x} \in \mathbb{Z}_+^n \\ \quad \mathbf{t} \in T \end{array} \right) < \pi_1 \tag{21}$$

whereas the master problem is identical to (5). We cannot utilize any (mixed) integer cut based on the \mathbf{t} -variables in the master problem, because the \mathbf{t} -variables are not present in the master problem. The ASLP we can use will therefore be equivalent to the ordinary ASLP (11). Note that (11) is a relaxation of the ASLP using $\mathbf{A}_2\mathbf{x} + \mathbf{B}\mathbf{t} = \mathbf{b}_2$. A consequence of this is that we cannot use (11) directly to prove optimality.

We can improve the ASLP in the following way for the special case where \mathbf{B} is a square matrix. Suppose that \mathbf{B} is invertible, then we can rewrite the constraint $\mathbf{A}_2\mathbf{x} + \mathbf{B}\mathbf{t} = \mathbf{b}_2$ in terms of \mathbf{t} as $\mathbf{t} = \mathbf{B}^{-1}(\mathbf{b}_2 - \mathbf{A}_2\mathbf{x})$. Furthermore, suppose that \mathbf{t} is bounded, i.e. $\boldsymbol{\alpha} \leq \mathbf{t} \leq \boldsymbol{\beta}$

⁶This is usual in resource constrained routing, where the pricing problems takes care of the resource constraints for the individual vehicles. Examples of such problems are given by Desaulniers et al. (1998) and Range (2006).

for some vectors α and β . If a component t_i is not lower bounded, we put $\alpha_i = -\infty$ and if it is not upper bounded, we put $\beta_i = \infty$. Thus

$$\begin{aligned} \alpha &\leq \mathbf{B}^{-1}(\mathbf{b}_2 - \mathbf{A}_2\mathbf{x}) \leq \beta \\ \Leftrightarrow \mathbf{B}^{-1}\mathbf{b}_2 - \beta &\leq \mathbf{B}^{-1}\mathbf{A}_2\mathbf{x} \leq \mathbf{B}^{-1}\mathbf{b}_2 - \alpha \end{aligned}$$

These additional constraints can now be added to the auxiliary SLP (11) and we obtain

$$\begin{aligned} z^*(T) = \min \quad & \sum_{j \in \mathcal{X}_0} x_j && + s_{lb} \\ \text{s.t.} \quad & \mathbf{c}^t \mathbf{x} && - s_{lb} = z_{lb} \\ & \mathbf{A}_0 \mathbf{x} && = \mathbf{b}_0 \\ & \mathbf{A}_1 \mathbf{x} && = \mathbf{b}_1 \\ & \mathbf{B}^{-1} \mathbf{A}_2 \mathbf{x} && \leq \mathbf{B}^{-1} \mathbf{b}_2 - \alpha \\ & \mathbf{B}^{-1} \mathbf{A}_2 \mathbf{x} && \geq \mathbf{B}^{-1} \mathbf{b}_2 - \beta \\ & \mathbf{d}^t \mathbf{x} + d_{lb}^t s_{lb} && \geq e_t, \quad t \in T \\ & \mathbf{x} \geq \mathbf{0}, \quad s_{lb} \geq 0 && \end{aligned} \tag{22}$$

The main weakness of this strengthening is that it requires a square \mathbf{B} -matrix, which is rare in formulations like (20). Hence other strengthenings are needed in practice.

When using (11) as ASLP, we implicitly capture some of the effects from $\mathbf{A}_2\mathbf{x} + \mathbf{B}\mathbf{t} = \mathbf{b}_2$. This is due to the lower-bound constraint $\mathbf{c}\mathbf{x} \geq z_{lp}$, which is based on the lower bound of the master problem. Each column in the master problem has taken the constraint $\mathbf{A}_2\mathbf{x} + \mathbf{B}\mathbf{t} = \mathbf{b}_2$ into account as it is a part of the pricing problem. Hence, only the combined effect of columns is not taken into account.

5 Future Research

The method for generating valid integer inequalities for the Dantzig-Wolfe decomposition remains to be tested in practice. We are currently initializing a computational study of our approach and will report the results in a future paper. In the study we will be interested in the integer inequalities generated from a basic feasible solution for an LP. We believe that the Gomory Fundamental Cuts (described above) and the Intersection Cuts (Balas, 1971) will enable us to tighten the lower bounds of the LP relaxation. The computational study may also reveal technical difficulties such as tailing off and numeric instability. The study will also show how much the lower bound is tightened by using our approach. The intuition is that our approach will not be as strong as hand-crafted problem dependent inequalities. Hence it should be used in combination with such inequalities (if any have been identified).

We have not extended our approach to MILP, but it seems to be a straightforward extension, which we leave for future research. The Dantzig-Wolfe decomposition for MILPs is reviewed by Vanderbeck (2005b), whereas a review of valid inequalities for MILP is given by Cornuéjols (2006). We believe that combining the Dantzig-Wolfe decomposition with general valid inequalities for MILP in the robust fashion as described above will improve the performance of the general purpose Column Generation.

6 Conclusion

In this paper, we have presented a general-purpose integer Cutting-Plane procedure for the Dantzig-Wolfe decomposition of integer linear programs. The method can use the general-purpose integer cutting planes which are used in State-of-the-art Branch-and-Cut solvers such as CPLEX. We believe that the addition of general-purpose integer cutting planes to Price-and-Cut algorithms will improve the lower bound obtained and therefore reduce the size of the Branch-and-Bound tree when using Branch-Price-and-Cut. This will, in turn, allow us to solve larger problem instances. We have also provided extensions to more general structures, which are commonly used in a range of applications. These applications include the Vehicle Routing Problem with Resource Constraints, the Elementary Shortest Path Problem with Resource Constraints, the Cutting Stock Problem, and Scheduling Problems. Most of these applications have a large portion of binary variables which may also be fixed on their upper bounds. We have argued that it may be easier to generate violated cuts for binary problems as they have a lesser degree of freedom in the ASLP.

The advantage of the procedure presented is that it can be applied in the solution of any ILP where the Dantzig-Wolfe decomposition is utilized. It will, furthermore, combine the information from the constraints in the master problems with the information from the constraints in the pricing problem and thereby it may identify violated inequalities, which are not identified by special-purpose hand-crafted inequalities. We have shown, by a small example, how a lower bound constraint can be used to obtain a stronger set of integer cuts, which may exclude some integer solutions, but never the optimal integer solution.

It is important to note that the Cutting-Plane procedure presented in this paper requires no modification of the pricing algorithms. This is due to the fact that we generate the integer cuts in terms of the original variables instead of the extremal set of variables. Hence, the procedure can easily be added to existing applications of Branch-Price-and-Cut.

All-in-all, we have presented a general robust procedure for generating integer cutting planes in Price-and-Cut which we believe will improve the general performance of the Branch-Price-and-Cut approach.

References

- Amor, H. B. & Carvalho, J. V. d. (2005). Cutting stock problems. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation*, number 5 in GERAD 25th Anniversary Series. Springer.
- Balas, E. (1971). Intersection cuts - a new type of cutting planes for integer programming. *Operations Research*, 19(1), 19–39.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46.
- Cornuéjols, G. (2006). Valid inequalities for mixed integer linear programs. To appear in *Mathematical Programming*, found at <http://integer.tepper.cmu.edu/webpub/CornuejolsRioMPS.ps>.

- Dantzig, G. B. & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8, 101–111.
- Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M. M., Soumis, F., & Villeneuve, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. G. Crainic & G. Laporte (Eds.), *Fleet Management and Logistics* (pp. 57–93). Kuwer Academic Publishers.
- Desaulniers, G., Desrosiers, J., & Solomon, M. M. (2005). *Column Generation*. Number 5 in GERAD 25th Anniversary Series. Springer.
- Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2).
- Garfinkel, R. S. & Nemhauser, G. L. (1972). *Integer Programming*. Wiley-Interscience.
- Gomory, R. E. (1963a). An algorithm for integer solutions to linear programs. In R. L. Graves & P. Wolfe (Eds.), *Recent Advances in Mathematical Programming* (pp. 269–302). McGraw-Hill.
- Gomory, R. E. (1963b). An all-integer integer programming algorithm. In J. F. Muth & G. L. Thompson (Eds.), *Industrial Scheduling*, Prentice-Hall International Series in Management (pp. 193–206). Prentice-Hall.
- Jepsen, M., Spoorendonk, S., Petersen, B., & Pisinger, D. (2006). A non-robust branch-and-cut-and-price algorithm for the vehicle routing problem with time windows. Technical Report 06/03, DIKU, Department of Computer Science, University of Copenhagen.
- Jünger, M., Reinelt, G., & Thienel, S. (1995). Practical problem solving with cutting plane algorithms in combinatorial optimization. In W. Cook, L. Lovász, & P. Seymour (Eds.), *Combinatorial optimization. Papers from the DIMACS special year. Papers from workshops held at DIMACS at Rutgers University*, volume 20 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (pp. 111–152).
- Nemhauser, G. & Wolsey, L. (1988). *Integer and Combinatorial Optimization*. Wiley Interscience Series in Discrete Mathematics and Optimization. Wiley.
- Nemhauser, G. L. & Park, S. (1991). A polyhedral approach to edge coloring. *Operations Research Letters*, 10(6), 315–322.
- Ralphs, T. K. & Galati, M. V. (2006). Decomposition and dynamic cut generation in integer linear programming. *Mathematical Programming*, 106(2), 261–285.
- Range, T. M. (2006). *Exact Solution of Resource Constrained Routing using Branch-Price-and-Cut*. PhD thesis, University of Southern Denmark, Faculty of Social Sciences, Department of Business & Economics.
- Vanderbeck, F. (1995). *Decomposition and Column Generation for Integer Programs*. PhD thesis, Universit Catholique de Louvain, Facult des Sciences Appliques, Unit de Gestion Industrielle.

- Vanderbeck, F. (2000). On dantzig-wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48, 111–128.
- Vanderbeck, F. (2005a). Branching in branch-and-price: a generic scheme. Technical Report U-05-14, University Bordeaux, Applied Mathematics.
- Vanderbeck, F. (2005b). Implementing mixed integer column generation. In G. Desaulniers, J. Desrosiers, & M. M. Solomon (Eds.), *Column Generation*, number 5 in GERAD 25th Anniversary Series. Springer.