# Decomposition and Dynamic Cut Generation in Integer Programming

T.K. Ralphs*  M.V. Galati†

Revised November 23, 2003

**Abstract**

Decomposition algorithms such as Lagrangian relaxation and Dantzig-Wolfe decomposition are well-known methods that can be used to develop bounds for integer programming problems. We draw connections between these classical approaches and techniques based on generating strong valid inequalities. We also discuss several methods for incorporating dynamic cut generation into traditional decomposition methods and present a new paradigm for separation called decompose and cut. The methods we discuss take advantage of the fact that separation of a solution to a combinatorial relaxation is often much easier than separation of an arbitrary real vector.

## 1 Introduction

In this paper, we consider methods for computing a bound on the value of an optimal solution to a given integer linear program (ILP)[1]. Computing bounds is an essential element of the branch and bound algorithm, which is the most effective and most commonly used method for solving such mathematical programs. Bounds are typically computed by solving a *bounding subproblem*, which is either a *relaxation* (a minimization problem with a feasible set that includes that of the original ILP) or a *dual* (a maximization problem whose optimal value is no larger than that of the original ILP) [37]. In what follows, we first review a standard paradigm, known as *decomposition*, for generating both relaxations and duals, and then discuss the relationships between several well-known decomposition methods. Finally, we present extensions of these basic methods that incorporate dynamic generation of valid inequalities, a standard technique for improving bounds in integer programming.

The effectiveness of the branch and bound algorithm depends largely on the *quality* of the bounds computed by the bounding procedure. We define this quality to be the size of the *relative gap*, which is the difference between the bound and the optimal solution value, as a fraction of the optimal solution value. The size of the relative gap is an indicator of

*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, `tkralphs@lehigh.edu`, `http://www.lehigh.edu/~tkr2`

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, `magh@lehigh.edu`, `http://sagan.ie.lehigh.edu/mgalati`

[1] We assume minimization throughout.

the effectiveness of the bounding procedure in the context of branch and bound. The most commonly used bounding subproblem is the linear programming (LP) relaxation obtained by dropping the integrality requirement from an ILP formulation. This LP relaxation is usually much easier to solve than the original ILP, but for many difficult ILPs, the resulting gap is too large to be effective in a branch and bound framework.

Another approach to formulating a bounding subproblem is to drop a subset of the inequalities from the ILP formulation to obtain a *combinatorial relaxation*. Although the resulting gap may still be too large, the bound can be improved by applying one of several traditional decomposition techniques, such as Dantzig-Wolfe decomposition [17] or Lagrangian relaxation [20, 13]. Computing a bound using these techniques requires the solution of multiple instances of the relaxation in an iterative fashion and so depends on the existence of an effective algorithm for solving the relaxation. However, the resulting bound may be much stronger than the bound yielded by the initial LP relaxation[2].

Although decomposition methods can yield improved bounds, one advantage of LP-based methods is their extensibility. Vast amounts of effort have gone into developing techniques for dynamically generating valid inequalities that can be used to strengthen the initial LP relaxation. This technique is known as the *cutting plane method* and when applied within a branch and bound framework, the approach is called *branch and cut* (see [14]). In traditional incarnations of both Lagrangian relaxation and Dantzig-Wolfe decomposition, one is forced to work with a single relaxation and this relaxation must be one that can be effectively solved. As we will see, this means that the bound cannot be easily improved beyond that yielded by the initial bounding subproblem. Using the cutting plane method, on the other hand, one can generate inequalities valid for any number of possible relaxations, as well as inequalities valid for the original feasible set. In fact, in a finite number of iterations, the cutting plane method can produce the *optimal* bound. Decomposition methods, however, provide valuable additional information about the structure of the solution that can aid in the generation of valid inequalities.

In the remainder of the paper, we propose a framework that strengthens the relationship between the cutting plane method and decomposition methods and allows the techniques to be combined in various ways. Using this framework, we describe modified versions of traditional decomposition algorithms, called *dynamic decomposition algorithms*, that are analogs of the cutting plane method. The extensions we describe increase the flexibility of techniques such as Lagrangian relaxation and Dantzig-Wolfe decomposition and allow the incorporation of dynamically generated valid inequalities to yield improved bounds. We also introduce methods that take advantage of the ability to separate the solutions of certain combinatorial relaxations from the convex hull of solutions to the original ILP. In many cases, this separation problem can be solved much more effectively than that for an arbitrary real vector. Finally, we illustrate the techniques with some examples. It should be noted that this paper presents only theory and methodology. The implementational details of these methods are quite involved and will be discussed in a forthcoming companion paper.

[2]We consider any linear program that is a relaxation of an ILP to be an *LP relaxation*. The word *initial* here separates the LP relaxation obtained from the original formulation of the ILP from those augmented by dynamically generated valid inequalities.

## 2 Definitions and Notation

To simplify the exposition, we consider only pure integer linear programming problems (PILPs) with finite upper and lower bounds on all variables. However, the framework can easily be extended to more general settings. For the remainder of the paper, we consider a specific PILP instance whose feasible set is the integer vectors contained in the polyhedron $\mathcal{Q} = \{x \in \mathbb{R}^n : Ax \geq b\}$, where $A \in \mathbb{Q}^{m \times n}$ is the constraint matrix and $b \in \mathbb{Q}^m$ is the right-hand-side vector. Let $\mathcal{F} = \mathcal{Q} \cap \mathbb{Z}^n$ be the set of all *feasible solutions* to the PILP and let the polyhedron $\mathcal{P}$ be the convex hull of $\mathcal{F}$.

We are interested in two related problems associated with the polyhedron $\mathcal{P}$. The *optimization problem* for $\mathcal{P}$ is that of determining

$$z_{IP} = \min_{x \in \mathcal{F}} \{c^\top x\} = \min_{x \in \mathcal{P}} \{c^\top x\} = \min_{x \in \mathbb{Z}^n} \{c^\top x : Ax \geq b\} \tag{1}$$

for a given *cost vector* $c \in \mathbb{Q}^n$. Here $z_{IP}$ is the *optimal value* and any $x^* \in \mathcal{F}$ such that $c^\top x^* = z_{IP}$ is an *optimal solution*. We refer to the process of solving this problem as that of *optimizing over* $\mathcal{P}$. A related problem is the *separation problem* for $\mathcal{P}$. Given $x \in \mathbb{R}^n$, the problem of separating $x$ from $\mathcal{P}$ is that of determining whether $x \in \mathcal{P}$ and if not, determining $a \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ such that $a^\top y \geq \beta \ \forall y \in \mathcal{P}$ but $a^\top x < \beta$. A pair $(a, \beta) \in \mathbb{R}^{n+1}$ such that $a^\top y \geq \beta \ \forall y \in \mathcal{P}$ is called a *valid inequality*[3] for $\mathcal{P}$ and is said to be *violated* by $x \in \mathbb{R}^n$ if $a^\top x < \beta$. In [25], it was shown that the separation problem for $\mathcal{P}$ is polynomially equivalent to the optimization problem for $\mathcal{P}$.

To apply the principle of decomposition, we consider the combinatorial relaxation

$$\min_{x \in \mathcal{F}'} \{c^\top x\} = \min_{x \in \mathcal{P}'} \{c^\top x\} = \min_{x \in \mathbb{Z}^n} \{c^\top x : A'x \geq b'\} \tag{2}$$

of (1) defined by the enlarged feasible set $\mathcal{F}' = \{x \in \mathbb{Z}^n : A'x \geq b'\}$, where $A' \in \mathbb{Q}^{m' \times n}, b' \in \mathbb{Q}^{m'}$. Note that in order for (2) to be a relaxation of (1), we must have $\mathcal{P} \subset \mathcal{P}' = conv(\mathcal{F}')$. As usual, we assume that there exists an effective[4] algorithm for optimizing over $\mathcal{P}'$. In addition, we assume that there exists an effective algorithm for separating an arbitrary real vector from $\mathcal{P}'$. This assumption is reasonable in light of the polynomial equivalence of optimization and separation.

Along with $\mathcal{P}'$ is associated a set of *side constraints*. Let $[A'', b''] \in \mathbb{Q}^{m'' \times n''}$ be a set of additional inequalities needed to describe $\mathcal{F}$, i.e., $[A'', b'']$ is such that $\mathcal{F} = \{x \in \mathbb{Z}^n : A'x \geq b', A''x \geq b''\}$. We denote by $\mathcal{Q}'$ the polyhedron described by the inequalities $[A', b']$ and by $\mathcal{Q}''$ the polyhedron described by the inequalities $[A'', b'']$. Hence, the initial LP relaxation is the linear program defined by $\mathcal{Q} = \mathcal{Q}' \cap \mathcal{Q}''$, and the *LP bound* is given by

$$z_{LP} = \min_{x \in \mathcal{Q}} \{c^\top x\} = \min_{x \in \mathbb{R}^n} \{c^\top x : A'x \geq b', A''x \geq b''\} \tag{3}$$

Note that traditionally, $[A', b']$ and $[A'', b'']$ are a partition of the rows of $[A, b]$ into a set of "nice constraints" and a set of "complicating constraints," so that the rows of $[A', b']$

[3]Note that we assume all inequalities are expressed in "$\geq$" form.

[4]The term *effective* here is not defined rigorously, but is a subjective measure denoting an algorithm that has a "reasonable" average-case running time.

and $[A'', b'']$ are among the rows of $[A, b]$. However, there are cases in which the relaxation we would like to consider is obtained simply by modifying the right hand side of certain inequalities (for example, see the discussion of the Vehicle Routing Problem in Section 5). In such cases, the rows of $[A', b']$ may not be present among the rows of $[A, b]$. In fact, the same row may be present in both $A'$ and $A''$ with different right-hand-sides.

# 3 Traditional Decomposition Methods

The goal of the decomposition approach is to improve on $z_{LP}$ by taking advantage of our ability to optimize over and/or separate from $\mathcal{P}'$. In this section, we briefly review the classical bounding methods that take this approach.

## 3.1 Lagrangian Relaxation

For a given vector $u \in \mathbb{R}_+^{m''}$, the *Lagrangian relaxation* of (1) is given by

$$z_{LR}(u) = \min_{s \in \mathcal{F}'} \{(c^\top - u^\top A'')s + u^\top b''\} \tag{4}$$

It is easily shown that $z_{LR}(u)$ is a lower bound on $z_{IP}$ for any $u \geq 0$. The elements of the vector $u$ are called *Lagrange multipliers* or *dual multipliers* with respect to the rows of $[A'', b'']$. The problem

$$z_{LD} = \max_{u \in \mathbb{R}_+^{m''}} \{z_{LR}(u)\}. \tag{5}$$

of maximizing this bound over all choices of dual multipliers is a dual to (1) called the *Lagrangian dual* and also provides a lower bound $z_{LD}$, which we call the *LD bound.* A vector of multipliers $\hat{u}$ that yield the largest bound are called *optimal (dual) multipliers.*

There are two basic approaches to solving a Lagrangian dual. First, as $z_{LR}(u)$ is a concave function of $u$, it can be maximized using subgradient optimization. Note that most of the computational effort goes into evaluating $z_{LR}(u)$ for a given $u$. This evaluation is an optimization problem over $\mathcal{P}'$, which we assumed could be solved effectively. The second method of solving (5) involves rewriting it as the equivalent linear program

$$z_{LD} \quad = \quad \max_{\alpha \in \mathbb{R}, u \in \mathbb{R}_+^{m''}} \{\alpha + u^\top b'' : \alpha \leq (c^\top - u^\top A'')s \ \forall s \in \mathcal{F}'\} \tag{6}$$

The number of inequalities in this linear program is $|\mathcal{F}'|$, so we cannot construct it explicitly. However, this program can be solved by the cutting plane method, since the separation problem is again an optimization problem over $\mathcal{P}'$. Both of these approaches are described in detail in [37].

## 3.2 Dantzig-Wolfe Decomposition

The approach of Dantzig-Wolfe decomposition is to reformulate (1) by implicitly requiring the solution to be a member of $\mathcal{F}'$, while explicitly enforcing the inequalities $[A'', b'']$.

Relaxing the integrality constraints of this reformulation, we obtain the linear program

$$z_{DW} = \min_{\lambda \in \mathbb{R}_+^{\mathcal{F}'}} \{c^\top (\sum_{s \in \mathcal{F}'} s\lambda_s) : A''(\sum_{s \in \mathcal{F}'} s\lambda_s) \geq b'', \sum_{s \in \mathcal{F}'} \lambda_s = 1\}, \tag{7}$$

which we call a *Dantzig-Wolfe LP*. Although the number of columns in this linear program is $|\mathcal{F}'|$, it can be solved by dynamic column generation, where the column-generation subproblem is again an optimization problem over $\mathcal{P}'$. For this reason, Dantzig-Wolfe is sometimes referred to generically as a *column-generation method*. Applying Dantzig-Wolfe decomposition within the context of branch and bound results in a method known as *branch and price* [11].

It is easy to verify that the Dantzig-Wolfe LP is the dual of (6), which immediately shows that $z_{DW} = z_{LD}$ (see [38] for a treatment of this fact). Hence, $z_{DW}$ is a valid lower bound on $z_{IP}$ that we call the *DW bound*. Note that the dual variables in the Dantzig-Wolfe LP correspond to the dual multipliers from Section 3.1. Furthermore, if we combine the members of $\mathcal{F}'$ using $\hat{\lambda}$, an optimal solution to the Dantzig-Wolfe LP, to obtain

$$\hat{x} = \sum_{s \in \mathcal{F}'} s\hat{\lambda}_s, \tag{8}$$

then we see that $z_{DW} = c^\top \hat{x}$. Since $\hat{x}$ must lie within $\mathcal{P}' \subseteq \mathcal{Q}'$ and also within $\mathcal{Q}''$, this shows that $z_{DW} \geq z_{LP}$.

### 3.3 Cutting Plane Method

In the cutting plane method, which is described in more detail in Section 4, inequalities describing $\mathcal{P}'$ (i.e., the facet-defining inequalities) are generated dynamically by separating the solutions to a series of LP relaxations from $\mathcal{P}'$. In this way, the initial LP relaxation is iteratively augmented to obtain the *CP bound*,

$$z_{CP} = \min_{x \in \mathcal{P}'} \{c^\top x : A''x \geq b''\}. \tag{9}$$

Note that $\hat{x}$, as defined in (8), is an optimal solution to this augmented linear program, which we call the *cutting plane LP*. Hence the CP bound is equal to both the DW bound and the LD bound. We refer to $\hat{x}$ as an *optimal fractional solution* and the augmented linear program above as the *cutting plane LP*.

### 3.4 A Common Framework

Summarizing what we have seen so far, the following well-known result of Geoffrion [23] relates the three methods just described.

**Theorem 1** $z_{IP} \geq c^\top \hat{x} = z_{LD} = z_{DW} = z_{CP} \geq z_{LP}$.

A graphical depiction of this bound is shown in Figure 1. Theorem 1 shows that the three approaches just described are simply three different algorithms for computing the same

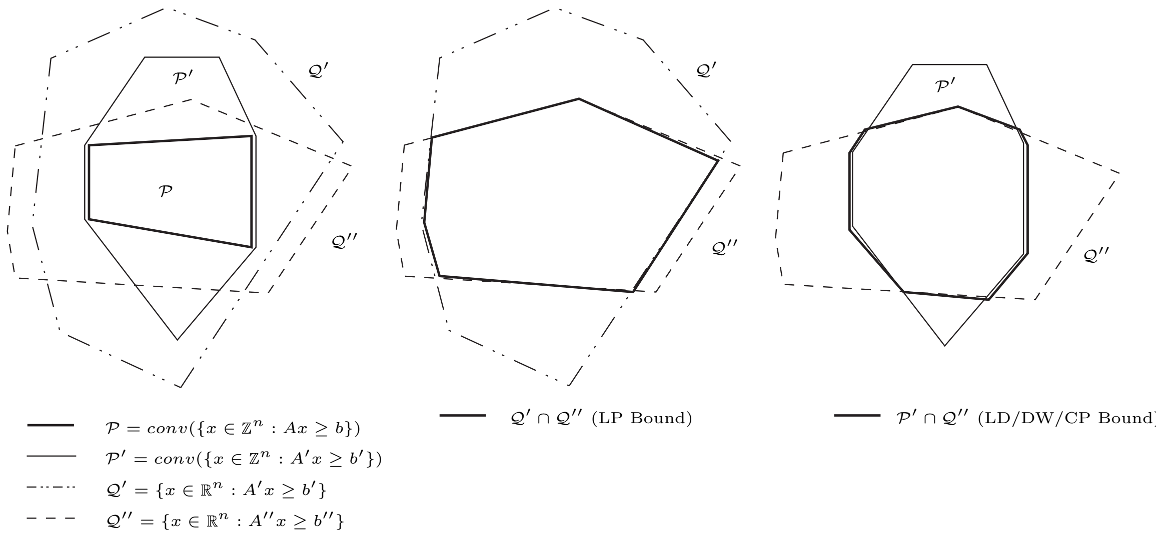



Figure 1: Illustration of the LP and LD/DW/CP bounds

quantity. In fact, the close relationship of these three methods can be made even more explicit. In each of the three methods, we are given a polyhedron $\mathcal{P}$ over which we would like to optimize, along with two additional polyhedra, denoted here by $\mathcal{Q}''$ and $\mathcal{P}'$, each of which contain $\mathcal{P}$. The polyhedron $\mathcal{Q}''$ has a small description that can be formulated explicitly, while the polyhedron $\mathcal{P}'$ has a much larger description and is represented implicitly, i.e., portions of the description are generated dynamically using our ability to effectively optimize/separate. To describe their roles in this framework, we call $\mathcal{P}$ the *original polyhedron*, $\mathcal{Q}''$ the *explicit polyhedron*, and $\mathcal{P}'$ the *implicit polyhedron*.

Recall that by the Weyl-Minkowski Theorem, every bounded rational polyhedron has two descriptions—one as the intersection of half-spaces (the *outer representation*) and one as the convex hull of its extreme points (the *inner representation*) [37]. The conceptual difference between Dantzig-Wolfe decomposition and the cutting plane method is that Dantzig-Wolfe decomposition utilizes an inner representation of $\mathcal{P}'$, generated dynamically by solving the corresponding optimization problem, whereas the cutting plane method relies on an outer representation of $\mathcal{P}'$, generated dynamically by solving the separation problem for $\mathcal{P}'$. In theory, we have the same choice of representation for the explicit polyhedron. It is intriguing to ponder the implications of this choice.

In this framework, the cutting plane method and column-generation methods for solving ILPs can also be viewed as decomposition methods. In the cutting plane method, the explicit polyhedron is simply the polyhedron described by the initial LP relaxation and the implicit polyhedron is any polyhedron containing $\mathcal{P}$ for which there exists an algorithm for solving the separation problem. On the other hand, most column-generation algorithms can be viewed as a form of Dantzig-Wolfe decomposition [43], with the implicit polyhedron being the convex hull of extremal solutions to the column-generation subproblem and the explicit polyhedron formed by the inequalities present in the master problem. Viewing these methods in a common framework yields a new and fruitful look at their relationship

**Cutting Plane Method**

Input: An instance of PILP.
Output: A lower bound $z$ on the optimal solution value for the instance.

1. Construct the initial LP relaxation $\text{LP}^0$ and set $i \leftarrow 0$.

   $z_{LP} = \min_{x \in \mathbb{R}^n}\{c^\top x : A'x \geq b', A''x \geq b''\}$

2. Solve $\text{LP}^i$ to obtain an optimal solution $\hat{x}^i$ and valid lower bound $z^i = c^\top \hat{x}^i$.

3. Attempt to separate $\hat{x}^i$ from $\mathcal{P}$, generating a set $[D^i, d^i]$ of valid inequalities violated by $\hat{x}^i$.

4. If valid inequalities were found in Step 3, form the augmented LP relaxation $\text{LP}^{i+1}$ by setting $[A'', b''] \leftarrow \left[\begin{smallmatrix} A'' & b'' \\ D^i & d^i \end{smallmatrix}\right]$. Then, set $i \leftarrow i+1$ and go to Step 2.

5. If no valid inequalities were found in Step 3, then output $z^i$.

Figure 2: Basic outline of the cutting plane method

to traditional decomposition methods, as discussed in the next section.

# 4 Dynamic Decomposition Methods

As we have already mentioned, one of the advantages of the cutting plane method over traditional decomposition approaches is the option of adding heuristically generated valid inequalities to the cutting plane LP. This can be thought of as a dynamic tightening of either the explicit or the implicit polyhedron. We focus here on the former interpretation and develop several *dynamic decomposition methods* using either a Dantzig-Wolfe LP or a Lagrangian dual as the bounding subproblem. Note that it is also possible to develop analogs of these methods based on the latter interpretation—the development is similar and is not be presented here. To begin, we first examine the basic steps of the cutting plane method, shown in Figure 2, assuming the problem at hand is feasible and bounded. The important step to note is Step 3, in which we try to tighten the explicit polyhedron by adding a new inequality valid for $\mathcal{P}$, but violated by $\hat{x}^i$.

In principle, a generalization of this procedure, which we generically call the *dynamic decomposition method*, can be used with either a Dantzig-Wolfe LP or a Lagrangian dual as the bounding subproblem. The steps are generalized as shown in Figure 3. Again, the important step is Step 3, generating a set of *improving inequalities*, i.e., inequalities valid for $\mathcal{P}$ that when added to the description of the explicit polyhedron result in an increase in the computed bound. Putting aside the important question of exactly how Step 3 is to be accomplished, the method is straightforward. Steps 1 and 2 are performed as in a traditional decomposition framework. Step 4 is accomplished by simply adding the newly generated inequalities to the list $[A'', b'']$ and reforming the appropriate bounding subproblem.

**Dynamic Decomposition Method**

Input: An instance of PILP.
Output: A lower bound $z$ on the optimal solution value for the instance.

1. Construct the initial bounding subproblem $\mathrm{P}^0$ and set $i \leftarrow 0$.

   $z_{CP} = \min_{x \in \mathcal{P}'} \{c^\top x : A''x \geq b''\}$
   $z_{LD} = \max_{u \in \mathbb{R}^n_+} \min_{x \in \mathcal{P}'} \{(c^\top - u^\top A'')x + u^\top b''\}$
   $z_{DW} = \min_{\lambda \in \mathbb{R}^{\mathcal{F}'}_+} \{c^\top(\sum_{s \in \mathcal{F}'} s\lambda_s) : A''(\sum_{s \in \mathcal{F}'} s\lambda_s) \geq b'', \sum_{s \in \mathcal{F}'} \lambda_s = 1\}$

2. Solve $\mathrm{P}^i$ to obtain a valid lower bound $z^i$.

3. Attempt to generate a set of improving inequalities $[D^i, d^i]$ valid for $\mathcal{P}$.

4. If valid inequalities were found in Step 3, form the bounding subproblem $\mathrm{P}^{i+1}$ by setting $[A'', b''] \leftarrow \left[\begin{smallmatrix} A'' & b'' \\ D^i & d^i \end{smallmatrix}\right]$. Then, set $i \leftarrow i+1$ and go to Step 2.

5. If no valid inequalities were found in Step 3, then output $z^i$.

Figure 3: Basic outline of the dynamic decomposition method

By changing the bounding subproblem, we arrive at several variants on this basic theme. When the bounding subproblem is a Dantzig-Wolfe LP, we call the resulting method *price and cut*. When employed in a branch and bound framework, the overall technique is called *branch, price, and cut*. This method has been studied by a number of authors [11, 44, 28, 10, 42] and is described in more detail in Section 4.1. When the bounding subproblem is a Lagrangian dual, we call the method *relax and cut*. When relax and cut is used as the bounding procedure in branch and bound framework, we call the overall method *branch, relax, and cut*. This method has also been studied previously by several authors (see [33] for a survey) and is described in more detail in Section 4.2. Finally, in Section 4.3, we introduce a variant of the cutting plane method that employs a decomposition-based separation procedure. We call this method *decompose and cut* and embed it within a branch and bound framework to obtain the method *branch, decompose, and cut*.

Having outlined a general framework, we now discuss exactly how the constraint generation step (Step 3), which is the crux of the method, is accomplished. In the context of the cutting plane method, Many classes of inequalities valid for PILPs are known [1], but there is no easily verifiable sufficient condition for an inequality to be *improving*. Given an optimal solution $\hat{x}$ to the cutting plane LP, we can apply one of the many known techniques to separate $\hat{x}$ from $\mathcal{P}$ in order to obtain a valid inequality violated by $\hat{x}$. Violation of $\hat{x}$ is a necessary condition for an inequality to be improving, and hence such an inequality is *likely* to be effective. However, unless the inequality separates the entire optimal face $F$ to the cutting plane LP, it will not be improving. Because we want to refer to these well-known results later in the paper, we state them formally as theorem and corollary without proof.

See [41] for a thorough treatment of the theory of linear programming that leads to this result.

**Theorem 2** *Let $F$ be the face of optimal solutions to the cutting plane LP. Then $(a, \beta) \in \mathbb{R}^{n+1}$ is an improving inequality if and only if $a^\top y < \beta$ for all $y \in F$.*

**Corollary 1** *If $(a, \beta) \in \mathbb{R}^{n+1}$ is an improving inequality and $\hat{x}$ is an optimal solution to the cutting plane LP, then $a^\top \hat{x} < \beta$.*

Fortunately, even in the case where $F$ is not separated in its entirety, the augmented cutting plane LP must have a different optimal solution, which in turn may be used to generate more potential improving inequalities. Since the condition of Theorem 2 is difficult to verify, one typically terminates the bounding procedure when increases resulting from additional inequalities become "too small." In the next two sections, we examine how improving inequalities can be generated when the bounding subproblem is either a Dantzig-Wolfe LP or a Lagrangian dual. Then we return to the cutting plane method to discuss how decomposition can be used directly to improve our ability to solve the separation problem.

### 4.1 Price and Cut

We first consider using a Dantzig-Wolfe LP as the bounding subproblem in the procedure of Figure 3. We call this general approach *price and cut* because it involves simultaneous generation of columns[5] (pricing) and valid inequalities (cutting) during solution of the bounding subproblem. Simultaneous generation of both columns and valid inequalities is difficult in general because the addition of valid inequalities tends to destroy the structure of the column-generation subproblem (for a discussion of this, see [42]). Having solved the Dantzig-Wolfe LP, however, we can easily recover an optimal solution to the cutting plane LP using (8) and try to generate improving inequalities as in the cutting plane method. The generation of those valid inequalities takes place in the original space and does not destroy the structure of the column-generation subproblem in the Dantzig-Wolfe LP. Hence, the approach enables dynamic generation of valid inequalities while still retaining the bound improvement and other advantages yielded by a Dantzig-Wolfe decomposition.

Note that the same bound improvement could also be realized in price and cut by adding the generated inequalities directly to the cutting plane LP, as described earlier. Despite the apparent symmetry, there is one fundamental difference between the cutting plane method and price and cut. An optimal solution to the Dantzig-Wolfe LP, which we shall henceforth refer to as an *optimal decomposition*, provides a decomposition of $\hat{x}$ from (8) into a convex combination of members of $\mathcal{F}'$. In particular, the weight on each member of $\mathcal{F}'$ in the combination is given by the optimal decomposition. We refer to elements of $\mathcal{F}'$ that have a positive weight in this combination as *members of the decomposition.* The following theorem shows how the decomposition can be used to derive an alternative necessary condition for an inequality to be improving.

[5]To be more precise, we should use the term *variables* here. The term "column generation" implies a fixed number of rows.

**Separation using the Optimal Decomposition**

Input: An optimal decomposition $\hat{\lambda}$.
Output: A set $[D, d]$ of potentially improving inequalities for $\mathcal{P}$.

1. Form the set $\mathcal{D} = \{s \in \mathcal{F}' : \hat{\lambda}_s > 0\}$.
2. For each $s \in \mathcal{D}$, attempt to separate $s$ from $\mathcal{P}$ to obtain a set $[D^s, d^s]$ of violated inequalities.
3. Let $[D, d]$ be composed of the inequalities found in Step 2 that are also violated by $\hat{x}$, so that $D\hat{x} < d$.
4. Return $[D, d]$ as the set of potentially improving inequalities.

Figure 4: Separation using the optimal decomposition

**Theorem 3** *If $(a, \beta) \in \mathbb{R}^{(n+1)}$ is an improving inequality, then there must exist an $s \in \mathcal{F}'$ with $\hat{\lambda}_s > 0$ such that $a^\top s < \beta$.*

**Proof.** Let $(a, \beta)$ be an improving inequality and $\hat{x} = \sum_{s \in \mathcal{F}'} s\hat{\lambda}_s$. Suppose that $a^\top s \geq \beta$ for all $s \in \mathcal{F}'$ with $\hat{\lambda}_s > 0$. Since $\sum_{s \in \mathcal{F}'} \hat{\lambda}_s = 1$, we have $a^\top(\sum_{s \in \mathcal{F}'} s\hat{\lambda}_s) \geq \beta$. Hence, $a^\top \hat{x} = a^\top(\sum_{s \in \mathcal{F}'} s\hat{\lambda}_s) \geq \beta$, which is a contradiction of Corollary 1. ∎

In other words, an inequality can be improving only if it is violated by at least one member of the decomposition. If $\mathcal{I}$ is the set of all improving inequalities, then the following corollary is a restatement of Theorem 3.

**Corollary 2** $\mathcal{I} \subseteq \mathcal{V} = \{(a, \beta) \in \mathbb{R}^{(n+1)} : a^\top s < \beta$ *for some* $s \in \mathcal{F}'$ *such that* $\hat{\lambda}_s > 0\}$

The importance of this result is that in some cases, it is much easier to separate members of $\mathcal{F}'$ from $\mathcal{P}$ than to separate arbitrary real vectors. In fact, it is easy to find polyhedra for which the problem of separating an arbitrary real vector is difficult, but the problem of separating a solution to a given combinatorial relaxation is easy. Some examples are discussed in Section 5. In Figure 4, we propose a new separation algorithm to be embedded in price and cut that takes advantage of this fact.

The running time of this procedure depends in part on the cardinality of the decomposition. Carathéodory's Theorem assures us that any optimal decomposition must have less than or equal to $\dim(\mathcal{P}) + 1$ members. Unfortunately, even if we limit our search to a particular known class of inequalities (see Section 5 for a discussion of inequality classes), the number of such inequalities violated by each member of $\mathcal{D}$ in Step 2 may be extremely large and many of them may not be violated by $\hat{x}$ (and thus cannot be improving). Designing this procedure to produce only inequalities that are likely to be violated by $\hat{x}$ can be difficult. Note that the set $\mathcal{V}$ of Corollary 2 can also be extremely large; unless we enumerate *every* inequality in the set $\mathcal{V}$, the procedure does not guarantee that an improving inequality will be found, even if one exists. Some details of the implementation of this scheme can be found in Section 5 and in [40]. Further details will be provided in a companion paper on the computational aspects of these methods.

Before moving on, we would like to draw some further connections between price and cut and the traditional cutting plane method that shed further light on the technique we have proposed. Solving the Dantzig-Wolfe LP generally results in an optimal fractional solution that is contained in a proper face of $\mathcal{P}'$. We can characterize precisely the face of $\mathcal{P}'$ in which $\hat{x}$ must be contained. Consider the set

$$\mathcal{S} = \{s \in \mathcal{F}' : (c^\top - \hat{u}^\top A'')s = (c^\top - \hat{u}^\top A'')\hat{s}\}, \tag{10}$$

where $\hat{s}$ is some member of the decomposition (and hence corresponds to a basic variable in an optimal solution to the Dantzig-Wolfe LP). The following theorem tells us that the set $\mathcal{S}$ must contain all members of the decomposition.

**Theorem 4** $\{s \in \mathcal{F}' : \hat{\lambda}_s > 0\} \subseteq \mathcal{S}$.

**Proof.** Let $\hat{\alpha} = z_{LD} - \hat{u}^\top b''$, so that $(\hat{\alpha}, \hat{u})$ is an optimal solution to the LP (6). Let $s \in \mathcal{F}'$ be such that $\hat{\lambda}_s > 0$. By complementary slackness, we know that $\hat{\lambda}_s(\hat{\alpha} - (c^\top - \hat{u}^\top A'')s) = 0$. Therefore, $\hat{\alpha} = (c^\top - \hat{u}^\top A'')s$. Since, $\hat{\alpha} = (c^\top - \hat{u}^\top A'')\hat{s}$, we conclude that $s \in \mathcal{S}$. ■

In fact, the proof of this theorem shows something stronger—that the set $\mathcal{S}$ is comprised exactly of those members of $\mathcal{F}'$ corresponding to columns of the Dantzig-Wolfe LP with reduced cost zero. This fact has important implications in the next section, where we use it to draw connections to another dynamic decomposition method. The following result follows from Theorem 4.

**Theorem 5** conv*(S) is a face of* $\mathcal{P}'$ *and contains* $\hat{x}$.

**Proof.** We first show that *conv*$(\mathcal{S})$ is a face of $\mathcal{P}'$. First, note that

$$(c^\top - \hat{u}^\top A'', (c^\top - \hat{u}^\top A'')\hat{s})$$

defines a valid inequality for $\mathcal{P}'$ since $\hat{s}$ was generated by optimizing over $\mathcal{P}'$ with the objective function $c^\top - \hat{u}^\top A''$. Thus, replacing $\mathcal{F}'$ by $\mathcal{P}'$ in the definition (10), we obtain a face $G$ of $\mathcal{P}'$ that contains $\mathcal{S}$. We claim that *conv*$(\mathcal{S}) = G$. Since $G$ is convex and contains $\mathcal{S}$, it also contains *conv*$(\mathcal{S})$, so we need to show that *conv*$(\mathcal{S})$ contains $G$. By construction, all extreme points of $\mathcal{P}'$ are members of $\mathcal{F}'$. Therefore, $\mathcal{S}$ must contain all extremal members of $G$ and the claim follows. Hence, *conv*$(\mathcal{S})$ is a face of $\mathcal{P}'$.

The fact that $\hat{x} \in$ *conv*$(S)$ follows from the fact that $\hat{x}$ is a convex combination of members of $\mathcal{S}$. ■

An important consequence of this result is contained in the following corollary, which tells us that the face of optimal solutions to the cutting plane LP is contained in *conv*$(\mathcal{S}) \cap \mathcal{Q}''$.

**Corollary 3** *If $F$ is the face of optimal solutions to the cutting plane LP, then* $F \subseteq \text{conv}(\mathcal{S}) \cap \mathcal{Q}''$.

**Proof.** Since $\hat{x}$ is an arbitrary member of $F$, the result follows. ■

Hence, the convex hull of the decomposition is a subset of *conv*($\mathcal{S}$) that contains $\hat{x}$ and can be thought of as a surrogate for the face of optimal solutions to the cutting plane LP. Combining this corollary with Theorem 2, we conclude that separation of $\mathcal{S}$ from $\mathcal{P}$ is a sufficient condition for an inequality to be improving. Although this sufficient condition is difficult to verify in practice, it does provide additional motivation for the method described in Figure 4.

As noted earlier, *conv*($\mathcal{S}$) is typically a proper face of $\mathcal{P}'$. It is possible, however, for $\hat{x}$ to be an inner point of $\mathcal{P}'$. The following result tells us that in this case, *conv*($\mathcal{S}$) $= \mathcal{P}'$ and so the decomposition yields little or no information.

**Theorem 6** *If $\hat{x}$ is an inner point of $\mathcal{P}'$, then* $\text{conv}(\mathcal{S}) = \mathcal{P}'$.

**Proof.** We prove the contrapositive. Suppose *conv*($\mathcal{S}$) is a proper face of $\mathcal{P}'$. Then there exists a facet-defining valid inequality $(a, \beta) \in \mathbb{R}^{n+1}$ such that *conv*($\mathcal{S}$) $\subseteq \{x \in \mathbb{R}^n : ax = \beta\}$. By Theorem 5, $\hat{x} \in$ *conv*($\mathcal{S}$) and $\hat{x}$ therefore cannot satisfy the definition of an inner point. ■

In this case, illustrated graphically in Figure 5(a), $z_{DW} = z_{LP}$ and Dantzig-Wolfe decomposition does not improve the bound. All columns of the Dantzig-Wolfe LP have reduced cost zero and any extremal member of $\mathcal{F}'$ could be made a member of the decomposition. The effectiveness of the procedure is reduced in this case—this is an indication that the chosen relaxation may be too weak. A necessary condition for an optimal fractional solution to be an inner point of $\mathcal{P}'$ is that the dual value of the convexity constraint in an optimal solution to the Dantzig-Wolfe LP be zero. If this condition arises, then caution should be observed. This result is further examined in the next section.

A second case of potential interest is when $F =$ *conv*($\mathcal{S}$) $\cap \mathcal{Q}''$, illustrated graphically in Figure 5(b). This condition can be detected by examining the objective function values of the members of the decomposition. If they are all equal, any member of the decomposition that is contained in $\mathcal{Q}''$ (if one exists) must be optimal for the original ILP, since it is feasible and has objective function value equal to $z_{LP}$. In this case, all constraints of the Dantzig-Wolfe LP *other than* the convexity constraint must have dual value zero, since removing them does not change the optimal solution value. The more typical case, in which $F$ is a proper subset of *conv*($\mathcal{S}$) $\cap \mathcal{Q}''$, is shown in Figure 5(c).

## 4.2 Relax and Cut

We now move on to discuss the dynamic generation of valid inequalities when the bounding subproblem is the Lagrangian dual, a method known as *relax and cut*. Recall again that there are two basic approaches to solving the Lagrangian dual, as covered in Section 3.1. Solving the Lagrangian dual as the linear program (6) is equivalent to solving the Dantzig-Wolfe LP, so the methods just discussed apply directly without modification. Hence, we assume from here on that the Lagrangian dual is solved in the form (5), using subgradient optimization.

Suppose we have solved the Lagrangian dual to obtain $\hat{u}$, a vector of optimal multipliers and $\hat{s} = \text{argmin}\{z_{LR}(\hat{u})\}$. As before, let $\hat{\lambda}$ be an optimal decomposition and $\hat{x} = \sum_{s \in \mathcal{F}'} s\hat{\lambda}_s$

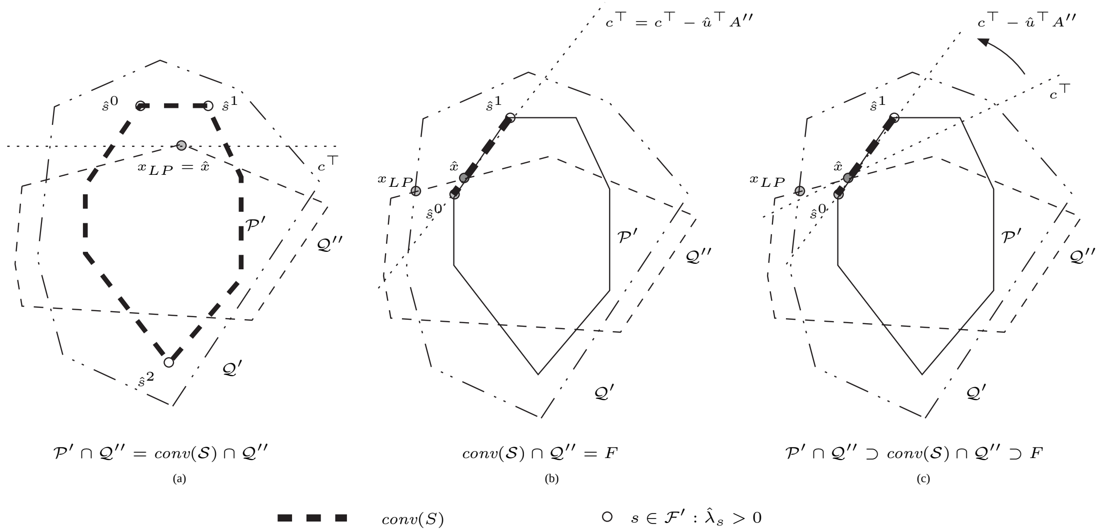



Figure 5: Illustration of bounds for different cost vectors

be an optimal fractional solution. The main consequence of using subgradient optimization to solve the Lagrangian dual is that we do not obtain the primal solution information available to us with both the cutting plane method and price and cut. This means there is no way of constructing an optimal fractional solution or verifying the condition of Corollary 1. We can, however, attempt to separate the solution to the Lagrangian relaxation (4), which is a member of $\mathcal{F}'$, from $\mathcal{P}$. Note that again, we are taking advantage of our ability to separate members of $\mathcal{F}'$ from $\mathcal{P}$ effectively. If successful, we immediately "dualize" this new constraint by adding it to $[A'', b'']$, as described in Section 4. This has the effect of introducing a new dual multiplier and slightly perturbing the objective function used to solve the Lagrangian relaxation.

As with both the cutting plane and price and cut methods, the difficulty with relax and cut is that the valid inequalities generated by separating $\hat{s}$ from $\mathcal{P}$ may not be improving, as Guignard first observed in [26]. As we have already noted, we cannot verify the condition of Corollary 1, which is the best available necessary condition for an inequality to be improving. To deepen our understanding of the potential effectiveness of the valid inequalities generated during relax and cut, we now further examine the relationship between $\hat{s}$ and $\hat{x}$.
From the reformulation of the Lagrangian dual as the linear program (6), we can see that each constraint binding at an optimal solution corresponds to an alternative optimal optimal solution to the Lagrangian subproblem with multipliers $\hat{u}$. The binding constraints of (6) correspond to variables with reduced cost zero in the Dantzig-Wolfe LP (7), so it follows immediately that the set of all alternative solutions to the Lagrangian subproblem with multipliers $\hat{u}$ is the set $\mathcal{S}$ from (10).

Because $\hat{x}$ is both an optimal solution to the cutting plane LP and is contained in $\mathcal{S}$, it also follows that

$$c^\top \hat{x} = (c^\top - \hat{u}^\top A'')\hat{x} + \hat{u}^\top b'',$$

In other words, the penalty term in the objective function of the Lagrangian subproblem (4) serves to rotate the original objective function so that it becomes parallel to the face $\mathcal{S}$, while constant term $\hat{u}^\top b''$ ensures that $\hat{x}$ has the same cost with both the original and the Lagrangian objective function. This is shown in Figure 5(c).

One conclusion that can be drawn from the above results is that the fundamental connection between relax and cut and price and cut is that solving the Dantzig-Wolfe LP produces a *set* of alternative optimal solutions to the Lagrangian Dual, at least one of which must be violated by a given improving inequality. This yields a verifiable necessary condition for a generated inequality to be improving. Relax and cut, on the other hand, produces only one member of this set, though possibly at a much lower computational cost. Even if improving inequalities exist, it is possible that none of them are violated by $\hat{s}$, especially if $\hat{s}$ has a small weight in the optimal decomposition. As in price and cut, when $\hat{x}$ is an inner point of $\mathcal{P}'$, the decomposition does not improve the bound and all members of $\mathcal{F}'$ are alternative optimal solutions to the Lagrangian subproblem dual multipliers $\hat{u}$. This situation is depicted in Figure 5(a). In this case, separating $\hat{s}$ is unlikely to yield an improving inequality.

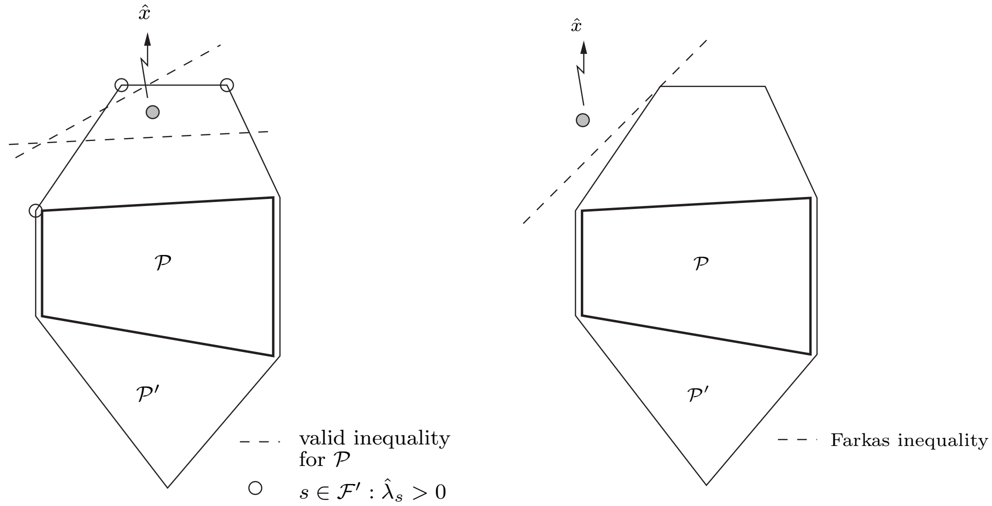
### 4.3 Decompose and Cut

The use of an optimal decomposition to aid in separation is easy to extend to a traditional branch and cut framework using a technique originally proposed in [40], which we call *decompose and cut*. Suppose now that we are given an optimal fractional solution $\hat{x}$ derived directly by solving the cutting plane LP. Suppose again that given $s \in \mathcal{F}'$, we can determine efficiently whether $s \in \mathcal{F}$ and if not, generate a valid inequality $(a, \beta)$ violated by $s$. We describe here a methodology for separating $\hat{x}$ from the polyhedron $\mathcal{P}$ by first *decomposing* $\hat{x}$ (i.e., expressing $\hat{x}$ as a convex combination of members of $\mathcal{F}'$) and then separating each member of the decomposition from $\mathcal{P}$, as described in Figure 4. When employed within a branch and bound framework, we call the resulting method *branch, decompose, and cut*.

The difficult step is finding the decomposition of $\hat{x}$. This can be accomplished by solving a linear program whose columns are the members of $\mathcal{F}'$, as described in Figure 7. This linear program is reminiscent of a Dantzig-Wolfe LP and in fact can be solved using an analogous column-generation scheme, as described in Figure 8. This scheme can be seen as the “inverse” of the method described in Section 4.1, since it begins with the fractional solution $\hat{x}$ and tries to compute an optimal decomposition, instead of the other way around.

As before, if such a decomposition exists, we can be assured by Carathéodory’s Theorem that only a small number of columns of $\mathcal{F}'$ are necessary to express it. This ensures us that we only have to apply the separation procedure to a small number members of $\mathcal{F}'$. However, as before, we may need to be very careful about how the valid inequalities violated by each member of $\mathcal{F}'$ are enumerated in order to ensure a high probability of finding one that is also violated by $\hat{x}$. Implementational details of this method will be contained in a separate forthcoming paper on the method.

Note that in contrast to price and cut, it is possible that $\hat{x} \notin \mathcal{P}$. This could occur, for instance, if exact separation methods for $\mathcal{P}'$ are too expensive to apply consistently. In this case, it is obviously not possible to find a decomposition in Step 2. The proof of




(a) $\hat{x} \in \mathcal{P}'$ (b) $\hat{x} \notin \mathcal{P}'$

Figure 6: Illustration of decompose and cut

**Separation in Decompose and Cut**

<u>Input:</u> $\hat{x} \in \mathbb{R}^n$
<u>Output:</u> A valid inequality for $\mathcal{P}$ violated by $\hat{x}$, if one is found.

1. Apply standard separation techniques to separate $\hat{x}$. If one of these returns a violated inequality, then STOP and output the violated inequality.

2. Otherwise, solve the linear program

$$\max_{\lambda \in \mathbb{R}_+^{\mathcal{F}'}} \{\mathbf{0}^\top \lambda : \sum_{s \in \mathcal{F}'} s\lambda_s = \hat{x}, \ \sum_{s \in \mathcal{F}'} \lambda_s = 1\}, \tag{11}$$

   as in Figure 8

3. The result of Step 2 is either (1) a subset $\mathcal{D}$ of members of $\mathcal{F}'$ participating in a convex combination of $\hat{x}$, or (2) a valid inequality $(a, \beta)$ for $\mathcal{P}$ that is violated by $\hat{x}$. In the first case, go to Step 4. In the second case, STOP and output the violated inequality.

4. Attempt to separate each member of $\mathcal{D}$ from $\mathcal{P}$. For each inequality violated by a member of $\mathcal{D}$, check whether it is also violated by $\hat{x}$. If an inequality violated by $\hat{x}$ is encountered, STOP and output it.

Figure 7: Separation in the decompose and cut method

**Column Generation in Decompose and Cut**

Input: $\hat{x} \in \mathbb{R}^n$
Output: Either (1) a valid inequality for $\mathcal{P}'$ violated by $\hat{x}$; or (2) a subset $\mathcal{D}$ of $\mathcal{F}'$ and a vector $\hat{\lambda} \in \mathbb{R}_+^D$ such that $\sum_{s \in D} \lambda_s s = \hat{x}$ and $\sum_{s \in D} \lambda_s = 1$.

2.0 Generate an initial subset $\mathcal{G}$ of $\mathcal{F}'$.

2.1 Solve (11) using the dual simplex algorithm, replacing $\mathcal{F}'$ by $\mathcal{G}$. If this linear program is feasible, then the elements of $\mathcal{F}'$ corresponding to the nonzero components of $\hat{\lambda}$, the current solution, comprise the set $\mathcal{D}$, so STOP.

2.2 Otherwise, let $r$ be the row in which the dual unboundedness condition was discovered, and let $(a, -\beta)$ be the $r^{th}$ row of the current basis inverse. Solve an optimization problem over $\mathcal{P}'$ with cost vector $a$ and let $s$ be the resulting solution. If the optimal value is less than $\beta$, then add $s$ to $\mathcal{G}$ and go to 2.1. Otherwise, $(a, \beta)$ is an inequality valid for $\mathcal{P}$ and violated by $\hat{x}$, so STOP.

Figure 8: Column generation for the decompose and cut method

infeasibility for the linear program (11), however, provides an inequality separating $\hat{x}$ from $\mathcal{P}'$ at no additional expense. Hence, even if we fail to find a decomposition, we still find an inequality valid for $\mathcal{P}$ and violated by $\hat{x}$.

Applying decompose and cut in every iteration as the sole means of separation is theoretically equivalent to price and cut. In practice, however, the decomposition is only computed when needed, i.e., when less expensive separation heuristics fail to separate the optimal fractional solution. This could give decompose and cut an edge in terms of computational efficiency. In other respects, the computations performed in each method are similar.

# 5 Examples

In this section, we discuss several examples to illustrate the application of the methods described in Section 4. In implementing these dynamic decomposition methods, it is useful to follow a separation paradigm called the *template paradigm*, which has become a standard approach to separation for ILPs. Instead of considering all valid inequalities at once, the template paradigm considers various (finite) subsets of valid inequalities, called *classes*, whose coefficients conform to the structure of a given *template*. The separation problem for a class of inequalities is then that of determining whether a given real vector lies in the polyhedron described by all inequalities in the class, and if not, determining an inequality from the class that is violated by the vector. See [16] for an excellent discussion of the template paradigm in integer programming.

In many cases, it is possible to solve the separation problem for a given class of inequalities valid for the polyhedron $\mathcal{P}$ effectively, though the general separation problem for $\mathcal{P}$ is difficult. We extend this framework by considering classes of valid inequalities along with combinatorial relaxations for which the separation of an arbitrary real vector is difficult but separation of solutions to the relaxation can be accomplished effectively. Any class of valid inequalities with this property is a candidate for separation using the methods we have described. We have discovered a number of common ILPs with classes of valid inequalities and relaxations that fit into this framework, such as the Orienteering Problem [19], The Edge-Weighted Clique Problem [27], Traveling Salesman Problem [6] and Cardinality Constrained Matching Problem [2]. In the following section, we describe three additional examples, the Vehicle Routing Problem, Axial Assignment Problem, and Steiner Tree Problem. For each problem, we discuss the key elements of the framework: (1) the original ILP formulation, (2) the explicit and implicit polyhedra used, and (3) a specific class or classes of valid inequalities that fits into the framework.

## 5.1 Vehicle Routing Problem

We first consider the *Vehicle Routing Problem* (VRP) introduced by Dantzig and Ramser [18]. In this $\mathcal{NP}$-hard optimization problem, a fleet of $k$ vehicles with uniform capacity $C$ must service known customer demands for a single commodity from a common depot at minimum cost. Let $V = \{1, \ldots, n\}$ index the set of customers and let the depot have index 0. Associated with each customer $i \in V$ is a demand $d_i$. The cost to travel from customer $i$ to $j$ is denoted $c_{ij}$ and we assume that $c_{ij} = c_{ji} > 0$ if $i \neq j$ and $c_{ii} = 0$.

By constructing an associated complete undirected graph $G$ with node set $N = V \cup \{0\}$ and edge set $E$, we can formulate the VRP as an integer program. A *route* is a set of nodes $R = \{i_1, i_2, \ldots, i_m\}$ such that the members of $R$ are distinct. The edge set of $R$ is $E_R = \{\{i_j, i_{j+1}\} : j \in 1, \ldots, m\}$, where $i_{m+1}$ is interpreted to be $i+1$. A feasible solution is then any subset of $E$ that is the union of the edge sets of $k$ disjoint routes $R_i, i \in [1..k]$, each satisfying the capacity restriction, i.e., $\sum_{j \in R_i} d_j \leq C$, $\forall i \in [1..k]$. Each route corresponds to a set of customers serviced by one of the $k$ vehicles. By associating a binary variable with each edge in the graph, we obtain the following formulation of this ILP [29]:

$$\min \sum_{e \in E} c_e x_e$$

$$\sum_{e \in \delta(0)} x_e = 2k, \tag{12}$$

$$\sum_{e \in \delta(i)} x_e = 2 \qquad \forall i \in V, \tag{13}$$

$$\sum_{e \in \delta(S)} x_e \geq 2b(S) \;\; \forall S \subseteq V,\ |S| > 1, \tag{14}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E(V), \tag{15}$$

$$x_e \in \{0, 1, 2\} \qquad \forall e \in \delta(0). \tag{16}$$

Here $b(S)$ represents a lower bound on the number of vehicles required to service the set of customers $S$. Inequalities (12) ensure that there are exactly $k$ vehicles, each departing from and returning to the depot, while inequalities (13) require that each customer must be serviced by exactly one vehicle. Inequalities (14), known as the *generalized subtour elimination constraints* (GSECs) can be viewed as a generalization of the subtour elimination constraints from the *Traveling Salesman Problem* (TSP) and enforce connectivity of the solution, as well as ensuring that no route has total demand exceeding capacity $C$. For ease of computation, we can define $b(S) = \left\lceil \left(\sum_{i \in S} d_i\right)/C \right\rceil$, a trivial lower bound on the number of vehicles required to service the set of customers $S$.

### 5.1.1 The Explicit Polyhedron and Valid Inequalities

Returning to our earlier notation and setup, the set of feasible solutions to the VRP is

$$\mathcal{F} = \{x \in \mathbb{R}^E : x \text{ satisfies } (12) - (16)\}$$

and we call $\mathcal{P} = conv(\mathcal{F})$ the *VRP polyhedron*. Many classes of valid inequalities for the VRP polyhedron have been reported in the literature (see [36] for a survey). However, for most known classes, the separation problem is intractable. The primary approach for overcoming this difficulty has been the development of *separation heuristics* that attempt to find a violated inequality of a given class, if one exists, but may fail.

The separation problem for GSECs (14) was shown to be in $\mathcal{NP}$-complete by Harche and Rinaldi (see [5]), even when $b(S)$ is taken to be $\left\lceil \left(\sum_{i \in S} d_i\right)/C \right\rceil$. Because there are an exponential number of GSECs, we cannot explicitly include them in the initial LP relaxation and they must be generated dynamically along with those from other classes described below. After dropping the inequalities (14), we define the explicit polyhedron to be

$$\mathcal{Q}'' = \{x \in \mathbb{R}^E : x \text{ satisfies } (12) - (13)\}.$$

In [15], Cornuéjols and Harche introduced a generalization of TSP comb inequalities valid for the VRP polyhedron. Araque et al. introduced another class of inequalities called *multistars* that are described in [4] and [30]. GSECs, comb inequalities, and multistars all induce facets of the VRP polyhedron under certain conditions. Next, we describe two relaxations that yield appropriate implicit polyhedra.

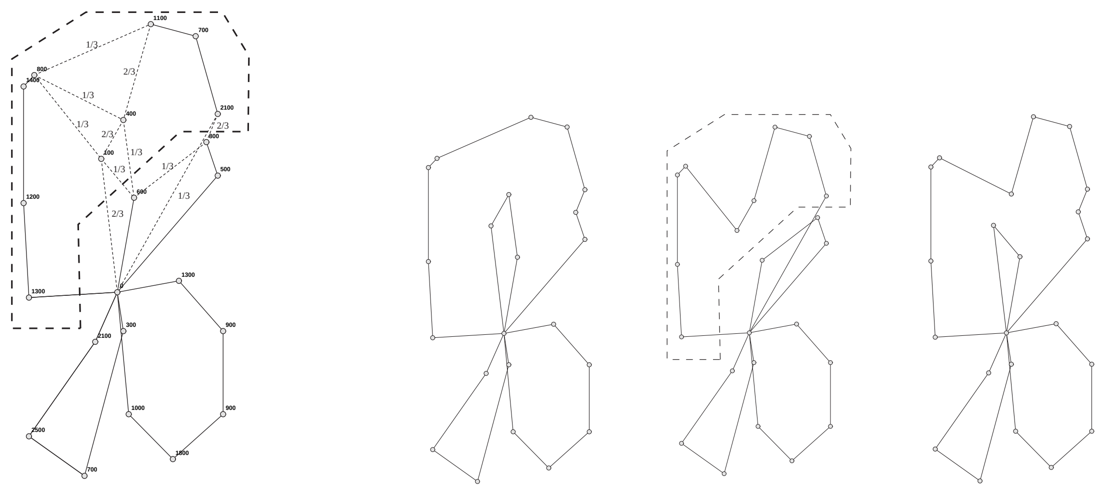
### 5.1.2 The Implicit Polyhedra and Separation

**Multiple Traveling Salesman Problem Relaxation.** The *Multiple Traveling Salesman Problem* (k-TSP) is an uncapacitated version of the VRP obtained by increasing the capacity to $C = \sum_{i \in S} d_i$. This gives $b(S) = 1$ for all $S \subseteq V$, and replaces the set of constraints (14) with

$$\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subseteq N,\ |S| > 1. \quad (17)$$

Our implicit polyhedron is then defined as $\mathcal{P}' = conv(\mathcal{F}')$, where

$$\mathcal{F}' = \{x \in \mathbb{R}^E : x \text{ satisfies } (12), (13), (15), (16), (17)\}.$$



(a) $\hat{x}$ (b) $\hat{\lambda}^1 = \frac{1}{3}$ (c) $\hat{\lambda}^2 = \frac{1}{3}$ (d) $\hat{\lambda}^3 = \frac{1}{3}$

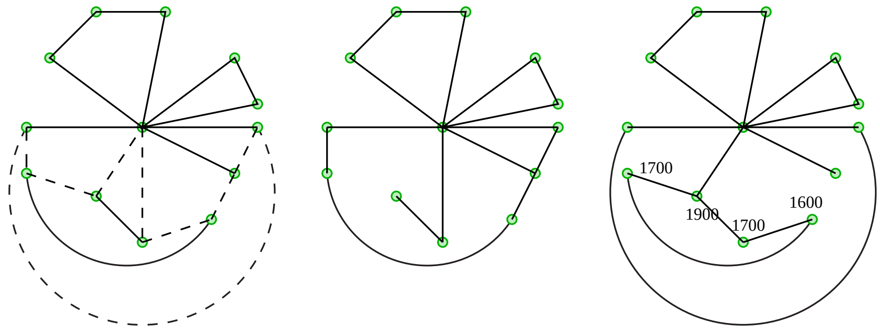
Figure 9: Example of a decomposition into k-TSP tours

Although the k-TSP is an $\mathcal{NP}$-hard optimization problem, small instances can be solved effectively by transformation into an equivalent TSP obtained by adjoining to the graph $k-1$ additional copies of node 0 and its incident edges.

Consider separating a member $s$ of $\mathcal{F}'$ from the polyhedron defined by all GSECs. We first construct the subgraph corresponding to $s$ with all edges incident to the depot vertex removed. We then find the connected components, which comprise the routes described earlier. In order to identify a GSEC that violates $s$, we simply compute the total demand of each route, checking whether it exceeds capacity. If not, the solution is feasible for the original ILP and does not violate any GSECs. If so, the set $S$ of customers in any route whose total demand exceeds capacity induces a violated GSEC. In this manner, we can separate a given member of $\mathcal{F}'$ from $\mathcal{P}$ with GSECs in $O(n)$ time. This separation routine can be used to generate GSECs with any of the dynamic decomposition methods previously described.

In [40], Ralphs et al. reports on the use of the branch, decompose and cut algorithm for separation of GSECs. Figure 9, reproduced from this paper, shows the decomposition of a fractional solution (a) to the VRP into three k-TSP tours (b,c,d). In this example, the capacity $C$ is 6000. By inspection we see that the top route of the second k-TSP tour (b) induces a violated GSEC. This constraint is also violated by the original fractional solution $\hat{x}$.

**k-Tree Relaxation.** A *k-Tree* is defined as a spanning subgraph of $G$ that has $n+k$ edges (recall that $G$ has $n+1$ nodes). A *degree-constrained k-Tree* ($k$-DCT), as defined by Fisher in [21], is a $k$-tree with degree $2k$ at node 0. The Minimum $k$-DCT Problem, is that of finding a minimum cost k-DCT, where the cost of a $k$-DCT is the sum of the costs on the edges present in the $k$-DCT. Since all feasible solutions to the VRP are $k$-DCTs with the same cost, the Minimum $k$-DCT Problem is a relaxation of the VRP. Fisher [21] introduced




(a) $\hat{x}$ (b) $\hat{\lambda}^1 = \frac{1}{2}$ (c) $\hat{\lambda} = \frac{1}{2}$

Figure 10: Example of a decomposition into $k$-DCTs

this relaxation as part of a Lagrangian relaxation-based algorithm for solving the VRP [21].

By first adding the redundant constraint

$$\sum_{e \in E} x_e = n + k \tag{18}$$

and then deleting the degree constraints (13), we obtain an integer programming formulation of the Minimum $k$-DCT Problem. We can hence define our implicit polyhedron as $\mathcal{P}' = conv(\mathcal{F}')$, where

$$\mathcal{F}' = \{x \in \mathbb{R}^E : x \text{ satisfies } (12), (15), (16), (17), (18)\}.$$

In [45], Wei and Yu give a polynomial algorithm for solving the Minimum $k$-DCT Problem that runs in $O(n^2 \log n)$ time.

Again consider separating a member $s$ of $\mathcal{F}'$ from the polyhedron defined by all GSECS. In [35], Martinhon et al. studied the use of the k-DCT relaxation for the VRP in the context of branch, relax, and cut. It is easy to see that for GSECs, an algorithm identical to that described above can be applied. Figure 10 gives an optimal fractional solution (a) to an LP relaxation of the VRP expressed as a convex combination of two k-DCTs (b,c). In this example, the capacity is again $C = 6000$ and by inspection we find a violated GSEC in the bottom component of the second k-Tree (c). Once again, this inequality is also violated by the optimal fractional solution. Martinhon et al. also give effective techniques for finding violated combs and multistars.

## 5.2 Axial Assignment Problem

The *Axial Assignment Problem* (AP3) is that of finding a minimum-weight clique cover of the complete tri-partite graph $K_{n,n,n}$. Let $I, J$ and $K$ be three disjoint sets with $|I| = |J| = |K| = n$ and set $H = I \times J \times K$. Then, AP3 can be formulated as the following binary integer program:

$$\min \sum_{(i,j,k)\in H} c_{ijk}x_{ijk}$$

$$\sum_{(j,k)\in J\times K} x_{ijk} = 1 \;\; \forall i \in I \tag{19}$$

$$\sum_{(i,k)\in I\times K} x_{ijk} = 1 \;\; \forall j \in J \tag{20}$$

$$\sum_{(i,j)\in I\times J} x_{ijk} = 1 \;\; \forall k \in K \tag{21}$$

$$x_{ijk} \in \{0,1\} \qquad \forall (i,j,k) \in H \tag{22}$$

A number of applications of AP3 can be found in the literature (see Piersjalla [18,19]). AP3 is known to be $\mathcal{NP}$-hard by reduction from the well known Set Partitioning Problem [22].

#### 5.2.1 The Explicit Polyhedron and Valid Inequalities

As before, we define the set of feasible solutions to AP3 as

$$\mathcal{F} = \{x \in \mathbb{R}^H : x \text{ satisfies } (19) - (22)\}$$

and set $\mathcal{P} = conv(\mathcal{F})$. In [8], Balas and Saltzman study the polyhedral structure of $\mathcal{P}$ and introduce several classes of facet-inducing inequalities.

Let $u, v \in H$ and define $|u \cap v|$ to be the numbers of components on which $u$ and $v$ agree. Let $C(u) = \{w \in H : |u \cap w| = 2\}$ and $C(u,v) = \{w \in H : |u \cap w| = 1, |w \cap v| = 2\}$. We consider two classes of facets $Q_1(u)$ and $P_1(u,v)$ for $\mathcal{P}$,

$$x_u + \sum_{w\in C(u)} x_w \leq 1 \;\; \forall u \in H, \tag{23}$$

$$x_u + \sum_{w\in C(u,v)} x_w \leq 1 \;\; \forall u, v \in H, |u \cap v| = 0. \tag{24}$$

Note that these include all the clique facets of the intersection graph of $K_{n,n,n}$ [8]. In [7], Balas and Qi describe algorithms for both $Q_1(u)$ and $P_1(u,v)$ that solve the separation problem in $O(n^3)$ time.

#### 5.2.2 The Implicit Polyhedron and Separation

Balas and Saltzman consider the use of the classical Assignment Problem (AP) as a relaxation of AP3 in an early implementation of branch, relax, and cut [9]. We follow their lead and define the implicit polyhedron to be $\mathcal{P}' = conv(\mathcal{F}')$, where

$$\mathcal{F}' = \{x \in \mathbb{R}^H : x \text{ satisfies } (20) - (22)\}.$$

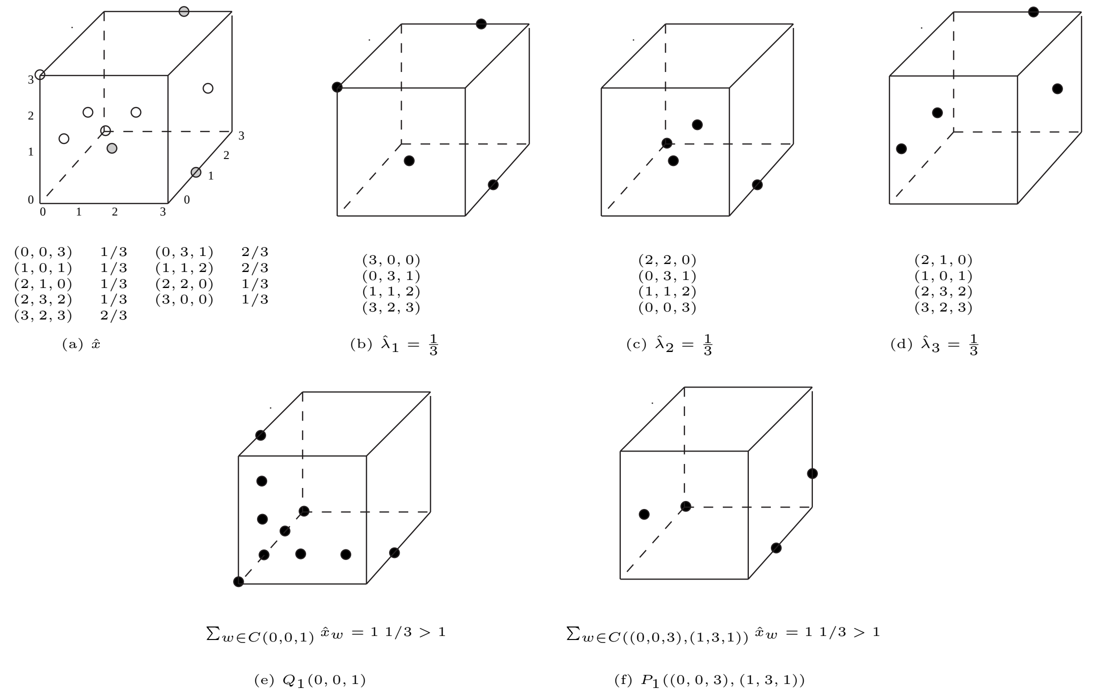



Figure 11: Example of a decomposition into assignments.

The AP can be solved in $O(n^{5/2} \log(nC))$ time where $C = \max_{w \in H} c_w$, by the cost-scaling algorithm [3]. Now consider separating a member $s$ of $\mathcal{F}'$ from the polyhedron defined by $Q_1(u)$ for all $u \in H$. Let $L(s)$ be the set of $n$ triplets corresponding to the nonzero components of $s$ (the assignment from $J$ to $K$). It is easy to see that if there exist $u, v \in L(s)$ such that $u = (i_0, j_0, k_0)$ and $v = (i_0, j_1, k_1)$, i.e., the assignment *overcovers* the set $I$, then both $Q(i_0, j_0, k_1)$ and $Q(i_0, j_1, k_0)$ are violated by $s$. Figure 11 shows the decomposition of an optimal fractional solution $\hat{x}$ (Figure 11(a)) into a convex combination of assignments (Figures 11(b-d)). In the second assignment (Figure 11(c)), the pair of triplets $(0, 3, 1)$ and $(0, 0, 3)$ satisfies the condition just discussed and identifies two violated valid inequalities, $Q_1(0, 3, 3)$ and $Q_1(0, 0, 1)$ which violate $\hat{s}$. The latter also violates $\hat{x}$ and is illustrated in Figure 11 (e). This separation routine runs in $O(n)$ time.

Now consider separating a member $s$ of $\mathcal{F}'$ from the polyhedron defined by $P_1(u, v)$ for all $u, v \in H$. As above, for any pair of assignments that correspond to nonzero components of $s$ and have the form $(i_0, j_0, k_0), (i_0, j_1, k_1)$, we know $s$ violates $P_1((i_0, j_0, k_0), (i, j_1, k_1))$, $\forall i \neq i_0$ and $P_1((i_0, j_1, k_1), (i, j_0, k_0)), \forall i \neq i_0$. In Figure 11, the second assignment (Figure 11(c)) is violated by $P_1((0, 0, 3), (1, 3, 1))$. This inequality is also violated by $\hat{x}$ and is illustrated in Figure 11 (f). Once again, this separation routine runs in $O(n)$ time.

### 5.3 Steiner Tree Problem

Let $G = (V, E)$ be a complete undirected graph with vertex set $V = \{1, ..., n\}$, edge set $E$ and a positive weight $c_{ij}$ associated with each edge $\{i, j\} \in E$. Let $T$ be a subset of $V$ called *terminals*. The *Steiner Tree Problem* (STP), which is in $\mathcal{NP}$-hard, is that of finding a subgraph that spans $T$ (called a *Steiner tree*) and has minimum cost, where cost is again defined as the sum of the costs of the edges present in the subgraph. In [12], Beasley formulated STP as a side constrained *Minimum Spanning Tree Problem* (MSTP) as follows. Assume without loss of generality, that $1 \in T$. Define the augmented graph $\bar{G} = (\bar{V}, \bar{E})$ where $\bar{V} = V \cup \{0\}$ and $\bar{E} = E \cup \{\{0, i\} : i \in V \setminus T \cup \{1\}\}$. Let $c_{0i} = 0$ for all $i \in V \setminus T \cup \{1\}$. Then, the STP is equivalent to finding a MST in $\bar{G}$ subject to the additional restriction that any vertex $i \in V \setminus T \cup \{1\}$ connected by the edge $\{0, i\} \in \bar{E}$ to vertex 0 must have degree one.

By associating a binary variable $x_{ij}$ with each edge $\{i, j\} \in \bar{E}$, indicating whether or not the edge is selected, we can then formulate the STP as the following integer program:

$$\min \sum_{i,j \in V} c_{ij} x_{ij}$$

$$\sum_{i,j \in \bar{V}} x_{ij} = n \tag{25}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \; \forall S \subseteq V \tag{26}$$

$$x_{0i} + x_{ij} \leq 1 \qquad \forall i \in V \setminus T, j \in V \setminus \{i\} \tag{27}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in \bar{V} \tag{28}$$

Inequalities (25) and (26) ensure that solution forms a spanning tree on $\bar{G}$. Inequalities (26) are subtour elimination constraints (similar to those used in the TSP). Inequalities (27) are the side constraints that ensure the solution can be converted to a Steiner tree by dropping the edges in $\bar{E} \setminus E$.

#### 5.3.1 The Explicit Polyhedron and Valid Inequalities

Returning again to our earlier notation, we now define the set of feasible solutions to the STP to be

$$\mathcal{F} = \{x \in \mathbb{R}^{\bar{E}} : x \text{ satisfies } (25) - (28)\}$$

and set $\mathcal{P} = conv(\mathcal{F})$. As with the VRP, we cannot include inequalities (26) in the description of the explicit polyhedron and hence define

$$\mathcal{Q}'' = \{x \in \mathbb{R}^{\bar{E}} : x \text{ satisfies } (25), (27)\}.$$

We consider two classes of valid inequalities that are lifted versions of the subtour elimination constraints.

$$\sum_{i,j \in S} x_{ij} + \sum_{i \in S \setminus T} x_{0i} \leq |S| - 1 \ \forall S \subseteq V, S \cap T \neq \emptyset, \tag{29}$$

$$\sum_{i,j \in S} x_{ij} + \sum_{i \in S \setminus \{v\}} x_{0i} \leq |S| - 1 \ \forall S \subseteq V, S \cap T \neq \emptyset, v \in S \tag{30}$$

The class of valid inequalities (29) were independently introduced by Goemans [24], Lucena [31] and Margot, Prodon, and Liebling [34], for another extended formulation of STP. The inequalities (30) were introduced in [24, 34]. The separation problem for inequalities (29) and (30) can be solved in $O(n^4)$ time through a series of max-flow computations.

### 5.3.2 The Implicit Polyhedron and Separation

In [32], Lucena considers the use of the *Minimum Spanning Tree Problem* (MSTP) as a relaxation of the STP in the context of a branch, relax, and cut algorithm. We therefore define our implicit polyhedron to be $\mathcal{P}' = conv(\mathcal{F}')$, where

$$\mathcal{F}' = \{x \in \mathbb{R}^{\bar{E}} : x \text{ satisfies } (25), (26), (28)\}$$

The MSTP can be solved in $O(n^2 \log n)$ time using the well-known algorithm of Prim [39]. Now consider the separation of a member of $s \in \mathcal{F}'$ from the polyhedron defined by the lifted subtour inequalities (29) and (30). In order to identify a violated inequality of the form (29) or (30) we simply traverse the tree from vertex 0 looking for an edge $\{0, i\}$ where $i \notin T$. This separation routine runs in $O(n)$.

## 6 Conclusions and Future Work

In this paper, we presented a unifying framework for incorporating dynamic cut generation into traditional decomposition methods. We have also introduced a new paradigm for the generation of improving inequalities based on decomposition and the separation of solutions to a combinatorial relaxation, a problem that is often much easier than that of separating arbitrary real vectors. Viewing the cutting plane method, Lagrangian relaxation, and Dantzig-Wolfe decomposition in a common algorithmic framework has yielded new insight into all three methods. The next step in this research is to complete a computational study that will allow practitioners to make intelligent choices between the three approaches. This study is already underway and the results are promising so far. As part of the study, we are implementing a generic framework that will allow users to implement these methods simply by providing a combinatorial relaxation, a solver for that relaxation, and separation routines for solutions to the relaxation. Such a framework will allow users access to a wide range of alternatives for solving integer programs using decomposition.

## References

[1] Aardal, K., and van Hoesel, S. Polyhedral techniques in combinatorial optimization, 1995.

[2] Aboudi, R., Hallefjord, A., and Jörnsten, K. A Facet Generation and Relaxation Technique Applied to an Assignment Problem with Side Constraints. *European Journal of Operational Research 50* (1991), 335–344.

[3] Ahuja, R., Magnanti, T., and Orlin, J. *Network Flows – Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[4] Araque, J., Kudva, G., Morin, T., and Pekny, J. A Branch and Cut Algorithm for Vehicle Routing Problems. *Annals of Operations Research 50* (1994), 37–59.

[5] Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Tech. rep., Université Joseph Fourier, Grenoble, France, 1995.

[6] Balas, E., and Christofides, N. A Restricted Lagrangian Approach to the Traveling Salesman Problem. *Mathematical Programming 21* (1981), 19–46.

[7] Balas, E., and Qi, L. Linear-time separation algorithms for the three-index assignment polytope. *Discrete Applied Mathematics 43* (1993), 1–12.

[8] Balas, E., and Saltzman, M. Facets of the three-index assignment polytope. *Discrete Applied Mathematics 23* (1989), 201–229.

[9] Balas, E., and Saltzman, M. An algorithm for the three-index assignment problem. *Operations Research 39* (1991), 150–161.

[10] Barnhart, C., Hane, C. A., and Vance, P. H. Using branch-and-price-and-cut to solve origin-destintation integer multi-commodity flow problems. *Operations Research 48(2)* (2000), 318–326.

[11] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. Branch and price: Column generation for solving huge integer programs. *Operations Research 46* (1998), 316–329.

[12] Beasley, J. A SST-Based Algorithm for the Steiner Problem in Graphs. *Networks 19* (1989), 1–16.

[13] Beasley, J. Lagrangean Relaxation. In *Modern Heuristic Techniques for Combinatorial Optimization*, C. Reeves, Ed. Wiley, 1993.

[14] Caprara, A., and Fischetti, M. Branch-and-cut algorithms. In *Annotated Bibliographies in Combinatorial Optimization* (1997), pp. 45–63.

[15] Cornuéjols, G., and Harche, F. Polyhedral Study of the Capacitated Vehicle Routing Problem. *Mathematical Programming 60* (1993), 21–52.

[16] D. Applegate and R. Bixby and V. Chvátal and W. Cook. TSP Cuts Which Do Not Conform to the Template Paradigm. In *Computational Combinatorial Optimization*. Springer, 2001, pp. 261–303.

[17] Dantzig, G., and Wolfe, P. Decomposition principle for linear programs. *Operations Research 8* (1960), 101–111.

[18] Danzig, G., and Ramser, R. The Truck Dispatching Problem. *Management Science 6* (1959), 80–91.

[19] Fischetti, M., González, J., and Toth, P. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing 10* (1998), 133–148.

[20] Fisher, M. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science 27* (1981).

[21] Fisher, M. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. Tech. rep., The Wharton School, University of Pennsylvania, Department of Decision Science, 1990.

[22] Garey, M. R., and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[23] Geoffrion, A. Lagrangian relaxation for integer programming. *Mathematical Programming Study 2* (December 1974), 82–114.

[24] Goemans, M. The Steiner Tree Polytope and Related Polyhedra. Tech. rep., MIT, Department of Mathematics, Cambridge, MA, 1991.

[25] Grötschel, M. and Lovász, L. and Schrijver, A. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica 1* (1981), 169.

[26] Guignard, M. Efficient cuts in Lagrangean "Relax-and-Cut" Schemes. *European Journal of Operational Research 105* (1998), 216–223.

[27] Hunting, M., Faigle, U., and Kern, W. A Lagrangian Relaxation Approach to the Edge-Weighted Clique Problem. *European Journal of Operational Research 131* (2001), 119–131.

[28] Kohl, N., Desrosiers, J., Madsen, O., Solomon, M., and Soumis, F. 2-path cuts for the Vehicle Routing Problem with Time Windows. *Transportation Science 33* (1999), 101–116.

[29] Laporte, G., Nobert, Y., and Desrouchers, M. Optimal routing with capacity and distance restrictions. *Operations Research 33* (1985), 1050.

[30] Letchford, A., Eglese, R., and Lysgaard, J. Mulitstars, partial multistars and the capacitated vehicle routing problem. *Mathematica Programming 94* (2002), 21–40.

[31] Lucena, A. Tight Bounds for the Steiner Problem in Graphs. Talk given at TIMS-XXX - SOBRAPO XXIII Joint International Meeting, Rio de Janeiro, July 1991.

[32] Lucena, A. Steiner Problem in Graphs: Lagrangean Relaxation and Cutting Planes. *COAL Bulletin 28* (1992), 2–8.

[33] Lucena, A. Relax and cut algorithms. Working paper, Departmaneto de Administracao, Universidade Federal do Rio de Janeiro, 2003.

[34] Margot, F., Prodon, A., and Liebling, T. Tree Polyhedron on 2-Tree. Tech. rep., École Polytechnique Fédérale de Lausanne, 1991.

[35] Martinhon, C., Lucena, A., and Maculan, N. Stronger K-Tree Relaxations for the Vehicle Routing Problem. unpublished working paper, December 2001.

[36] Naddef, D., and Rinaldi, G. Vehicle Routing. In *Branch and Cut*. SIAM, 2000.

[37] Nemhauser, G., and Wolsey, L. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.

[38] Parker, R., and Rardin, R. *Discrete Optimization*. Academic Press, Inc., 1988.

[39] Prim, R. Shortest Connection Networks and Some Generalizations. *Bell System Technical Journal 36* (1957), 1389–1401.

[40] Ralphs, T., Kopman, L., Pulleyblank, W., and Trotter Jr., L. On the Capacitated Vehicle Routing Problem. *Mathematical Programming 94* (2003), 343.

[41] V. Chvátal. *Linear Programming*. W.H. Freeman and Company, 1983.

[42] van den Akker, J., Hurkens, C., and Savelsbergh, M. Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing 12(2)* (2000), 111–124.

[43] Vanderbeck, F. *Decomposition and Column Generation for Integer Programs*. PhD thesis, Universite Catholiqué de Louvain, 1994.

[44] Vanderbeck, F. Lot-Sizing with Start-up Times. *Management Science 44* (1998), 1409–1425.

[45] Wei, G., and Yu, G. An Improved $O(n^2 \log n)$ Algorithm for the Degree-Constrained Minimum K-Tree Problem. Tech. rep., The University of Texas at Austin, Department of Management Sciences, 1995.