Decomposition in Integer Linear Programming

T.K. Ralphs* M.V. Galati†

Revised August 16, 2005

Abstract

Both cutting plane methods and traditional decomposition methods are procedures that compute a bound on the optimal value of an integer linear program (ILP) by constructing an approximation to the convex hull of feasible solutions. This approximation is obtained by intersecting the polyhedron associated with the continuous relaxation, which has an explicit representation, with an implicitly defined polyhedron having a description of exponential size. In this paper, we first review these classical procedures and then introduce a new class of bounding methods called *integrated decomposition methods*, in which the bound yielded by traditional approaches is potentially improved by introducing a second implicitly defined polyhedron. We also discuss the concept of *structured separation*, which is related to the well-known template paradigm for dynamically generating valid inequalities and is central to our algorithmic framework. Finally, we briefly introduce a software framework for implementing the methods discussed in the paper and illustrate the concepts through the presentation of applications.

1 Introduction

In this paper, we discuss the principle of decomposition as it applies to the computation of bounds on the value of an optimal solution to an integer linear program (ILP). Most bounding procedures for ILP are based on the generation of a polyhedron that approximates $\mathcal{P}$, the convex hull of feasible solutions. Solving an optimization problem over such a polyhedral approximation, provided it fully contains $\mathcal{P}$, produces a bound that can be used to drive a branch and bound algorithm. The effectiveness of the bounding procedure depends largely on how well $\mathcal{P}$ can be approximated. The most straightforward approximation is the *continuous approximation*, consisting simply of the linear constraints present in the original ILP formulation. The bound resulting from this approximation is frequently too weak to be effective, however. In such cases, it can be improved by dynamically generating additional polyhedral information that can be used to augment the approximation.

Traditional dynamic procedures for augmenting the continuous approximation can be grouped roughly into two categories. *Cutting plane methods* improve the approximation by dynamically generating half-spaces containing $\mathcal{P}$, i.e., valid inequalities, to form a second

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

†Analytical Solutions - Operations R & D, SAS Institute, Cary, NC 27513 `matthew.galati@sas.com`, `http://sagan.ie.lehigh.edu/mgalati`

polyhedron, and then intersect this second polyhedron with the continuous approximation to yield a final approximating polyhedron. With this approach, the valid inequalities are generated by solution of an associated *separation problem*. Generally, the addition of each valid inequality reduces the hypervolume of the approximating polyhedron, resulting in a potentially improved bound. Because they dynamically generate part of the description of the final approximating polyhedron as the intersection of half-spaces (an *outer representation*), we refer to cutting plane methods as *outer approximation methods*.

Traditional decomposition methods, on the other hand, improve the approximation by dynamically generating the extreme points of a polyhedron containing $\mathcal{P}$, which is again intersected with the continuous approximation, as in the cutting plane method, to yield a final approximating polyhedron. In this case, each successive extreme point is generated by solution of an associated *optimization problem* and at each step, the hypervolume of the approximating polyhedron is increased. Because decomposition methods dynamically generate part of the description of the approximating polyhedron as the convex hull of a finite set (an *inner representation*), we refer to these methods as *inner approximation methods*.

Both inner and outer methods work roughly by alternating between a procedure for computing solution and bound information (the *master problem*) and a procedure for augmenting the current approximation (the *subproblem*). The two approaches, however, differ in important ways. Outer methods require that the master problem produce "primal" solution information, which then becomes the input to the subproblem, a *separation problem*. Inner methods require "dual" solution information, which is then used as the input to the subproblem, an *optimization problem*. In this sense, the two approaches can be seen as "dual" to one another. A more important difference, however, is that the valid inequalities generated by an inner method can be valid with respect to *any* polyhedron containing $\mathcal{P}$ (see Section 5), whereas the extreme points generated by an inner method must ostensibly be from a single polyhedron. Procedures for generating new valid inequalities can also take advantage of knowledge of previously generated valid inequalities to further improve the approximation, whereas with inner methods, such "backward-looking" procedures do not appear to be possible. Finally, the separation procedures used in the cutting plane method can be heuristic in nature as long as it can be proven that the resulting half-spaces do actually contain $\mathcal{P}$. Although heuristic methods can be employed in solving the optimization problem required of an inner method, valid bounds are only obtained when using exact optimization. On the whole, outer methods have proven to be more flexible and powerful and this is reflected in their position as the approach of choice for solving most ILPs.

As we will show, however, inner methods do still have an important role to play. Although inner and outer methods have traditionally been considered separate and distinct, it is possible, in principle, to integrate them in a straightforward way. By doing so, we obtain bounds at least as good as those yielded by either approach alone. In such an integrated method, one alternates between a master problem that produces both primal and dual information, and either one of two subproblems, one an optimization problem and the other a separation problem. This may result in significant synergy between the subproblems, as information generated by solving the optimization subproblem can be used to generate cutting planes and vice versa.

The remainder of the paper is organized as follows. In Section 2, we introduce definitions and notation. In Section 3, we describe the principle of decomposition and its application to integer linear programming in a traditional setting. In Section 4, we extend the traditional framework to show how the cutting plane method can be integrated with either the Dantzig-Wolfe method or the Lagrangian method to yield improved bounds. In Section 5, we discuss solution of the separation subproblem and introduce an extension of the well-known template paradigm, called *structured separation*, inspired by the fact that separation of structured solutions is frequently easier than separation of arbitrary real vectors. We also introduce a decomposition-based separation algorithm called *decompose and cut* that exploits structured separation. In Section 6, we discuss some of the algorithms that can be used to solve the master problem. In Section 7, we describe a software framework for implementing the algorithms presented in the paper. Finally, in Section 8, we present applications that illustrate the principles discussed herein.

2 Definitions and Notation

For ease of exposition, we consider only pure integer linear programs with bounded, nonempty feasible regions, although the methods presented herein can be extended to more general settings. For the remainder of the paper, we consider an ILP whose feasible set is the integer vectors contained in the polyhedron $\mathcal{Q} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$, where $A \in \mathbb{Q}^{m \times n}$ is the constraint matrix and $b \in \mathbb{Q}^m$ is the vector of requirements. Let $\mathcal{F} = \mathcal{Q} \cap \mathbb{Z}^n$ be the feasible set and let $\mathcal{P}$ be the convex hull of $\mathcal{F}$. The canonical *optimization problem* for $\mathcal{P}$ is that of determining

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

for a given cost vector $c \in \mathbb{Q}^n$, where $z_{IP} = \infty$ if $\mathcal{F}$ is empty. We refer to such an ILP by the notation $ILP(\mathcal{P}, c)$. In what follows, we also consider the equivalent decision version of this problem, which is to determine, for a given upper bound U, whether there is a member of $\mathcal{P}$ with objective function value strictly better than U. We denote by $OPT(\mathcal{P}, c, U)$ a subroutine for solving this decision problem. The subroutine is assumed to return either the empty set, or a set of one *or more* (depending on the situation) members of $\mathcal{P}$ with objective value better than U.

A related problem is the *separation problem* for $\mathcal{P}$, which is typically already stated as a decision problem. Given $x \in \mathbb{R}^n$, the problem of separating x from $\mathcal{P}$ is that of deciding 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 a *valid inequality* for $\mathcal{P}$ and is said to be *violated* by $x \in \mathbb{R}^n$ if $a^\top x < \beta$. We denote by $SEP(\mathcal{P}, x)$ a subroutine that separates an arbitrary vector $x \in \mathbb{R}^n$ from polyhedron $\mathcal{P}$, returning either the empty set or a set of one or more violated valid inequalities. Note that the optimization form of the separation problem is that of finding the *most violated* inequality and is equivalent to the decision form stated here.

A closely related problem is the *facet identification problem*, which restricts the generated inequalities to only those that are *facet-defining* for $\mathcal{P}$. In [32], it was shown that the facet identification problem for $\mathcal{P}$ is polynomially equivalent to the optimization problem for $\mathcal{P}$ (in the worst case sense). However, a theme that arises in what follows is that the complexity of optimization and separation can vary significantly if either the input or the output must have known structure. If the solution to an optimization problem is required to be integer, the problem generally becomes much harder to solve. On the other hand, if the input vector to a separation problem is an integral vector, then the separation problem frequently becomes much *easier* to solve in the worst case. From the dual point of view, if the input cost vector of an optimization problem has known structure, such as being integral, this may make the problem easier. Requiring the output of the separation problem to have known structure is known as the template paradigm and may also make the separation problem easier, but such a requirement is essentially equivalent to enlarging $\mathcal{P}$. These concepts are discussed in more detail in Section 5.

3 The Principle of Decomposition

We now formalize some of the notions described in the introduction. Implementing a branch and bound algorithm for solving an ILP requires a procedure that will generate a lower bound as close as possible to the optimal value z_{IP}. The most commonly used method of bounding is to solve the linear programming (LP) relaxation obtained by removing the integrality requirement from the ILP formulation. The *LP Bound* is given by

$$z_{LP} = \min_{x \in \mathbb{R}^n} \{c^\top x \mid Ax \geq b\} = \min_{x \in \mathcal{Q}} \{c^\top x\}, \tag{2}$$

and is obtained by solving a linear program with the original objective function c over the polyhedron $\mathcal{Q}$. It is clear that $z_{LP} \leq z_{IP}$ since $\mathcal{P} \subseteq \mathcal{Q}$. This LP relaxation is usually much easier to solve than the original ILP, but z_{LP} may be arbitrarily far away from z_{IP} in general, so we need to consider more effective procedures.

In most cases, the description of $\mathcal{Q}$ is small enough that it can be represented explicitly and the bound computed using a standard linear programming algorithm. To improve the LP bound, decomposition methods construct a second approximating polyhedron that can be intersected with $\mathcal{Q}$ to form a better approximation. Unlike $\mathcal{Q}$, this second polyhedron usually has a description of exponential size, and we must generate portions of its description dynamically. Such a dynamic procedure is the basis for both cutting plane methods, which generate an outer approximation, and for traditional decomposition methods, such as the Dantzig-Wolfe method [19] and the Lagrangian method [22, 14], which generate inner approximations.

For the remainder of this section, we consider the relaxation of (1) defined by

$$\min_{x \in \mathbb{Z}^n} \{c^\top x \mid A'x \geq b'\} = \min_{x \in \mathcal{F}'} \{c^\top x\} = \min_{x \in \mathcal{P}'} \{c^\top x\}, \tag{3}$$

where $\mathcal{F} \subset \mathcal{F}' = \{x \in \mathbb{Z}^n \mid A'x \geq b'\}$ for some $A' \in \mathbb{Q}^{m' \times n}, b' \in \mathbb{Q}^{m'}$ and $\mathcal{P}'$ is the convex hull of $\mathcal{F}'$. Along with $\mathcal{P}'$ is associated a set of *side constraints* $[A'', b''] \in \mathbb{Q}^{m'' \times (n+1)}$ such that $\mathcal{Q} = \{x \in \mathbb{R}^n \mid A'x \geq b', A''x \geq b''\}$. We denote by $\mathcal{Q}'$ the polyhedron described by the inequalities $[A', b']$ and by $\mathcal{Q}''$ the polyhedron described by the inequalities $[A'', b'']$. Thus, $\mathcal{Q} = \mathcal{Q}' \cap \mathcal{Q}''$ and $\mathcal{F} = \{x \in \mathbb{Z}^n \mid x \in \mathcal{P}' \cap \mathcal{Q}''\}$. For the decomposition to be effective, we must have that $\mathcal{P}' \cap \mathcal{Q}'' \subset \mathcal{Q}$, so that the bound obtained by optimizing over $\mathcal{P}' \cap \mathcal{Q}''$ is at least as good as the LP bound and strictly better for some objective functions.

The description of $\mathcal{Q}''$ must also be "small" so that we can construct it explicitly. Finally, we assume that there exists an *effective* algorithm for optimizing over $\mathcal{P}'$ and thereby, for separating arbitrary real vectors from $\mathcal{P}'$. We are deliberately using the term *effective* here to denote an algorithm that has an acceptable average-case running time, since this is more relevant than worst-case behavior in our computational framework.

Traditional decomposition methods can all be viewed as techniques for iteratively computing the bound

$$z_D = \min_{x \in \mathcal{P}'} \{c^\top x \mid A''x \geq b''\} = \min_{x \in \mathcal{F}' \cap \mathcal{Q}''} \{c^\top x\} = \min_{x \in \mathcal{P}' \cap \mathcal{Q}''} \{c^\top x\}. \tag{4}$$

In Sections 3.1–3.3 below, we review the cutting plane method, the Dantzig-Wolfe method, and the Lagrangian method, all classical approaches that can be used to compute this bound. The common perspective motivates Section 4, where we consider a new class of decomposition methods called *integrated decomposition methods*, in which both inner *and* outer approximation techniques are used in tandem. In both this section and the next, we describe the methods at a high level and leave until later sections the discussion of how the master problem and subproblems are solved. To illustrate the effect of applying the decomposition principle, we now introduce two examples that we build on throughout the paper. The first is a simple generic ILP.

Example 1 Let the following be the formulation of a given ILP:

$$\begin{aligned}
\min \quad & x_1, \\
7x_1 - x_2 &\geq 13, && (5)\\
x_2 &\geq 1, && (6)\\
-x_1 + x_2 &\geq -3, && (7)\\
-4x_1 - x_2 &\geq -27, && (8)\\
-x_2 &\geq -5, && (9)\\
0.2x_1 - x_2 &\geq -4, && (10)\\
-x_1 - x_2 &\geq -8, && (11)\\
-0.4x_1 + x_2 &\geq 0.3, && (12)\\
x_1 + x_2 &\geq 4.5, && (13)\\
3x_1 + x_2 &\geq 9.5, && (14)\\
0.25x_1 - x_2 &\geq -3, && (15)\\
x &\in \mathbb{Z}^2. && (16)
\end{aligned}$$

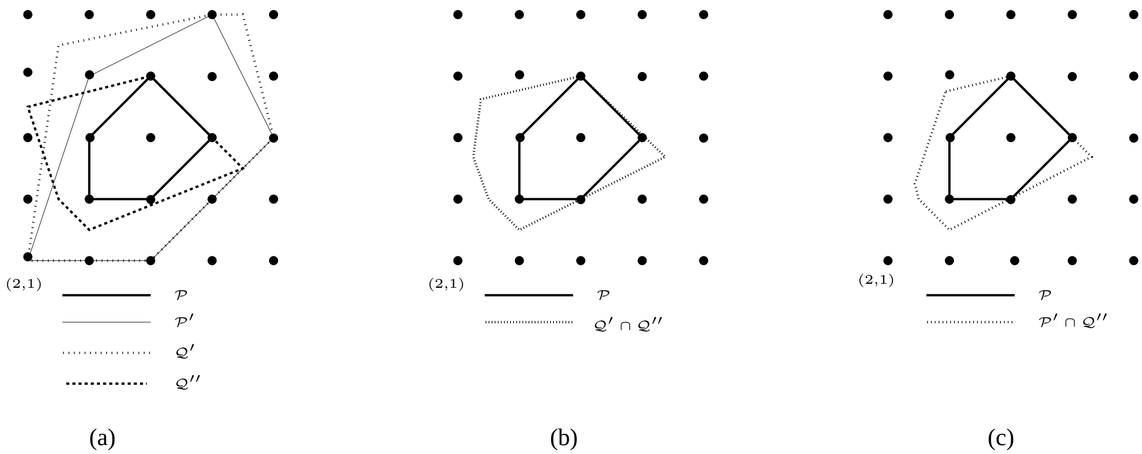


Figure 1: Polyhedra (Example 1)

In this example, we let

$$\begin{aligned} \mathcal{P} &= \operatorname{conv}\{x \in \mathbb{R}^2 \mid x \text{ satisfies } (5)-(16)\}, \\ \mathcal{Q}' &= \{x \in \mathbb{R}^2 \mid x \text{ satisfies } (5)-(10)\}, \\ \mathcal{Q}'' &= \{x \in \mathbb{R}^2 \mid x \text{ satisfies } (11)-(15)\}, \text{ and} \\ \mathcal{P}' &= \operatorname{conv}(\mathcal{Q}' \cap \mathbb{Z}^2). \end{aligned}$$

In Figure 1(a), we show the associated polyhedra, where the set of feasible solutions $\mathcal{F} = \mathcal{Q}' \cap \mathcal{Q}'' \cap \mathbb{Z}^2 = \mathcal{P}' \cap \mathcal{Q}'' \cap \mathbb{Z}^2$ and $\mathcal{P} = \operatorname{conv}(\mathcal{F})$. Figure 1(b) depicts the continuous approximation $\mathcal{Q}' \cap \mathcal{Q}''$, while Figure 1(c) shows the improved approximation $\mathcal{P}' \cap \mathcal{Q}''$. For the objective function in this example, optimization over $\mathcal{P}' \cap \mathcal{Q}''$ leads to an improvement over the LP bound obtained by optimization over $\mathcal{Q}$. ■

In our second example, we consider the classical *Traveling Salesman Problem* (TSP), a well-known combinatorial optimization problem. The TSP is in the complexity class $\mathcal{NP}$-hard, but lends itself well to the application of the principle of decomposition, as the standard formulation contains an exponential number of constraints and has a number of well-solved combinatorial relaxations.

Example 2 The Traveling Salesman Problem is that of finding a minimum cost *tour* in an undirected graph G with vertex set $V = \{0, 1, ..., |V| - 1\}$ and edge set E. We assume without loss of generality that G is complete. A tour is a connected subgraph for which each node has degree 2. The TSP is then to find such a subgraph of minimum cost, where the cost is the sum of the costs of the edges comprising the subgraph. With each edge $e \in E$, we therefore associate a binary variable x_e, indicating whether edge e is part of the subgraph, and a cost $c_e \in \mathbb{R}$. Let $\delta(S) = \{\{i,j\} \in E \mid i \in S, j \notin S\}, E(S:T) = \{\{i,j\} \mid i \in S, j \in T\}, E(S) = E(S:S)$ and $x(F) = \sum_{e \in F} x_e$. Then an ILP formulation of the TSP is as follows:

$$
\begin{aligned}
\min \sum_{e \in E} c_e x_e,& \\
x(\delta(\{i\})) &= 2 \quad && \forall i \in V, && (17)\\
x(E(S)) &\leq |S| - 1 && \forall S \subset V,\ 3 \leq |S| \leq |V| - 1, && (18)\\
0 \leq x_e &\leq 1 && \forall e \in E, && (19)\\
x_e &\in \mathbb{Z} && \forall e \in E. && (20)
\end{aligned}
$$

The continuous approximation, referred to as the *TSP polyhedron*, is then

$$\mathcal{P} = \text{conv}\{x \in \mathbb{R}^E \mid x \text{ satisfies } (17) - (20)\}.$$

The equations (17) are the *degree constraints*, which ensure that each vertex has degree two in the subgraph, while the inequalities (18) are known as the *subtour elimination constraints* (SECs) and enforce connectivity. Since there are an exponential number of SECs, it is impossible to explicitly construct the LP relaxation of TSP for large graphs. Following the pioneering work of Held and Karp [35], however, we can apply the principle of decomposition by employing the well-known *Minimum* 1-*Tree Problem*, a combinatorial relaxation of TSP.

A 1-tree is a tree spanning $V \setminus \{0\}$ plus two edges incident to vertex 0. A 1-tree is hence a subgraph containing exactly one cycle through vertex 0. The Minimum 1-Tree Problem is to find a 1-tree of minimum cost and can thus be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e, \\
x(\delta(\{0\})) &= 2, && (21)\\
x(E(V \setminus \{0\})) &= |V| - 2, && (22)\\
x(E(S)) &\leq |S| - 1 \quad \forall S \subset V \setminus \{0\}, 3 \leq |S| \leq |V| - 1, && (23)\\
x_e &\in \{0, 1\} \quad \forall e \in E. && (24)
\end{aligned}
$$

A minimum cost 1-tree can be obtained easily as the union of a minimum cost spanning tree of $V \setminus \{0\}$ plus two cheapest edges incident to vertex 0. For this example, we thus let $\mathcal{P}' = \text{conv}(\{x \in \mathbb{R}^E \mid x \text{ satisfies } (21) - (24)\})$ be the 1-Tree Polyhedron, while the degree and bound constraints comprise the polyhedron $\mathcal{Q}'' = \{x \in \mathbb{R}^E \mid x \text{ satisfies } (17) \text{ and } (19)\}$ and $\mathcal{Q}' = \{x \in \mathbb{R}^E \mid x \text{ satisfies } (18)\}$. Note that the bound constraints appear in the descriptions of both polyhedra for computational convenience. The set of feasible solutions to TSP is then $\mathcal{F} = \mathcal{P}' \cap \mathcal{Q}'' \cap \mathbb{Z}^E$. ■

3.1 Cutting Plane Method

Using the cutting plane method, the bound z_D can be obtained by dynamically generating portions of an outer description of $\mathcal{P}'$. Let $[D, d]$ denote the set of facet-defining inequalities of $\mathcal{P}'$, so that

$$\mathcal{P}' = \{x \in \mathbb{R}^n \mid Dx \geq d\}. \tag{25}$$

Cutting Plane Method

Input: An instance $ILP(\mathcal{P}, c)$.
Output: A lower bound z_{CP} on the optimal solution value for the instance, and $\hat{x}_{CP} \in \mathbb{R}^n$ such that $z_{CP} = c^\top \hat{x}_{CP}$.

1. **Initialize**: Construct an initial outer approximation

$$\mathcal{P}_O^0 = \{x \in \mathbb{R}^n \mid D^0 x \geq d^0\} \supseteq \mathcal{P}, \tag{27}$$

   where $D^0 = A''$ and $d^0 = b''$, and set $t \leftarrow 0$.

2. **Master Problem**: Solve the linear program

$$z_{CP}^t = \min_{x \in \mathbb{R}^n} \{c^\top x \mid D^t x \geq d^t\} \tag{28}$$

   to obtain the optimal value $z_{CP}^t = \min_{x \in \mathcal{P}_O^t} \{c^\top x\} \leq z_{IP}$ and optimal primal solution x_{CP}^t.

3. **Subproblem**: Call the subroutine $SEP(\mathcal{P}, x_{CP}^t)$ to generate a set of potentially *improving* valid inequalities $[\tilde{D}, \tilde{d}]$ for $\mathcal{P}$, violated by x_{CP}^t.

4. **Update**: If violated inequalities were found in Step 3, set $[D^{t+1}, d^{t+1}] \leftarrow \left[\begin{smallmatrix} D^t & d^t \\ \tilde{D} & \tilde{d} \end{smallmatrix}\right]$ to form a new outer approximation

$$\mathcal{P}_O^{t+1} = \{x \in \mathbb{R}^n \mid D^{t+1} x \leq d^{t+1}\} \supseteq \mathcal{P}, \tag{29}$$

   and set $t \leftarrow t + 1$. Go to Step 2.

5. If no violated inequalities were found, output $z_{CP} = z_{CP}^t \leq z_{IP}$ and $\hat{x}_{CP} = x_{CP}^t$.

Figure 2: Outline of the cutting plane method

Then the cutting plane formulation for the problem of calculating z_D can be written as

$$z_{CP} = \min_{x \in \mathcal{Q}''} \{c^\top x \mid Dx \geq d\}. \tag{26}$$

This is a linear program, but since the set of valid inequalities $[D, d]$ is potentially of exponential size, we dynamically generate them by solving a separation problem. An outline of the method is presented in Figure 2.

In Step 2, the master problem is a linear program whose feasible region is the current outer approximation $\mathcal{P}_O^t$, defined by a set of initial valid inequalities plus those generated dynamically in Step 3. Solving the master problem in iteration t, we generate the relaxed (primal) solution x_{CP}^t and a valid lower bound. In the figure, the initial set of inequalities is taken to be those of $\mathcal{Q}''$, since it is assumed that the facet-defining inequalities for $\mathcal{P}'$, which dominate those of $\mathcal{Q}'$, can be generated dynamically. In practice, however, this initial set may be chosen to include those of $\mathcal{Q}'$ or some other polyhedron, on an empirical basis.

In Step 3, we solve the subproblem, which is to generate a set of *improving* valid inequalities, i.e., valid inequalities that improve the bound when added to the current approximation. This step is usually accomplished by applying one of the many known techniques for separating x^t_{CP} from $\mathcal{P}$. The algorithmic details of the generation of valid inequalities are covered more thoroughly in Section 5, so the unfamiliar reader may wish to refer to this section for background or to [1] for a complete survey of techniques. It is well known that violation of x^t_{CP} is a necessary condition for an inequality to be improving, and hence, we generally use this condition to judge the potential effectiveness of generated valid inequalities. However, this condition is not sufficient and unless the inequality separates the entire optimal face of $\mathcal{P}^t_O$, it will not actually be improving. Because we want to refer to these results later in the paper, we state them formally as theorem and corollary without proof. See [59] for a thorough treatment of the theory of linear programming that leads to this result.

Theorem 1 *Let F be the face of optimal solutions to an LP over a nonempty, bounded polyhedron X with objective function vector f. Then (a, β) is an improving inequality for X with respect to f, i.e.,*

$$\min\{f^\top x \mid x \in X, a^\top x \geq \beta\} > \min\{f^\top x \mid x \in X\}, \tag{30}$$

if and only if $a^\top y < \beta$ for all $y \in F$.

Corollary 1 *If (a, β) is an improving inequality for X with respect to f, then $a^\top \hat{x} < \beta$, where $\hat{x}$ is any optimal solution to the linear program over X with objective function vector f.*

Even in the case when the optimal face cannot be 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 1 is difficult to verify, one typically terminates the bounding procedure when increases resulting from additional inequalities become "too small."

If we start with the continuous approximation $\mathcal{P}^0_O = \mathcal{Q}''$ and generate only facet-defining inequalities of $\mathcal{P}'$ in Step 3, then the procedure described here terminates in a finite number of steps with the bound $z_{CP} = z_D$ (see [52]). Since $\mathcal{P}^t_O \supseteq \mathcal{P}' \cap \mathcal{Q}'' \supseteq \mathcal{P}$, each step yields an approximation for $\mathcal{P}$, along with a valid bound. In Step 3, we are permitted to generate any valid inequality for $\mathcal{P}$, however, not just those that are facet-defining for $\mathcal{P}'$. In theory, this means that the cutting plane method can be used to compute the bound z_{IP} exactly. However, this is rarely practical.

To illustrate the cutting plane method, we show how it could be applied to generate the bound z_D for the ILPs of Examples 1 and 2. Since we are discussing the computation of the bound z_D, we only generate facet-defining inequalities for $\mathcal{P}'$ in these examples. We discuss more general scenarios later in the paper.

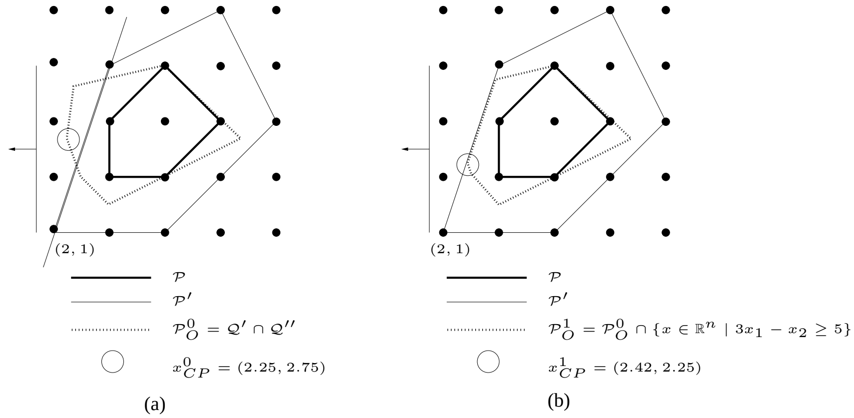


Figure 3: Cutting plane method (Example 1)

Example 1 (Continued) We define the initial outer approximation to be $\mathcal{P}_O^0 = \mathcal{Q}' \cap \mathcal{Q}'' = \{x \in \mathbb{R}^2 \mid x \text{ satisfies } (5) - (15)\}$, the continuous approximation.

Iteration 0: Solving the master problem over $\mathcal{P}_O^0$, we find an optimal primal solution $x_{CP}^0 = (2.25, 2.75)$ with bound $z_{CP}^0 = 2.25$, as shown in Figure 3(a). We then call the subroutine $SEP(\mathcal{P}, x_{CP}^0)$, generating facet-defining inequalities of $\mathcal{P}'$ that are violated by x_{CP}^0. One such facet-defining inequality, $3x_1 - x_2 \geq 5$, is pictured in Figure 3(a). We add this inequality to form a new outer approximation $\mathcal{P}_O^1$.

Iteration 1: We again solve the master problem, this time over $\mathcal{P}_O^1$, to find an optimal primal solution $x_{CP}^1 = (2.42, 2.25)$ and bound $z_{CP}^1 = 2.42$, as shown in Figure 3(b). We then call the subroutine $SEP(\mathcal{P}, x_{CP}^1)$. However, as illustrated in Figure 3(b), there are no more facet-defining inequalities violated by x_{CP}^1. In fact, further improvement in the bound would necessitate the addition of valid inequalities violated by points in $\mathcal{P}'$. Since we are only generating facets of $\mathcal{P}'$ in this example, the method terminates with bound $z_{CP} = 2.42 = z_D$. ■

We now consider the use of the cutting plane method for generating the bound z_D for the TSP of Example 2. Once again, we only generate facet-defining inequalities for $\mathcal{P}'$, the 1-tree polyhedron.

Example 2 (Continued) We define the initial outer approximation to be comprised of the degree constraints and the bound constraints, so that

$$\mathcal{P}_O^0 = \mathcal{Q}'' = \{x \in \mathbb{R}^E \mid x \text{ satisfies (17) and (19)}\}.$$

The bound z_D is then obtained by optimizing over the intersection of the 1-tree polyhedron with the polyhedron $\mathcal{Q}''$ defined by constraints (17) and (19). Note that because the 1-tree polyhedron has integer extreme points, we have that $z_D = z_{LP}$ in this case. To calculate z_D, however, we must dynamically generate violated facet-defining inequalities (the SECs (23))

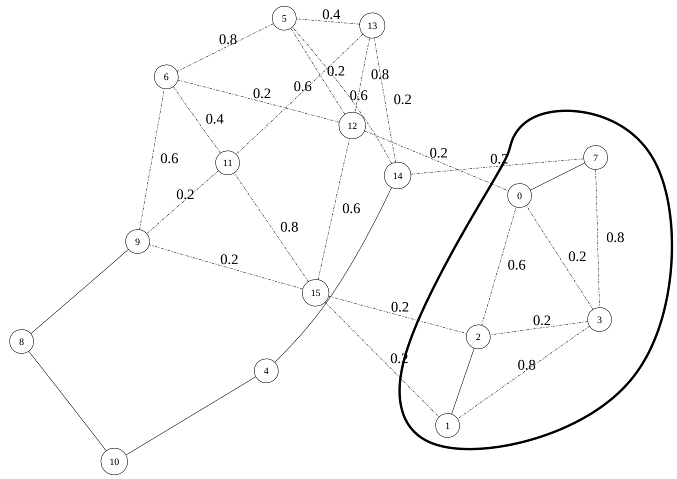


Figure 4: Finding violated inequalities in the cutting plane method (Example 2)

of the 1-tree polyhedron $\mathcal{P}'$ defined earlier. Given a vector $\hat{x} \in \mathbb{R}^E$ satisfying (17) and (19), the problem of finding an inequality of the form (23) violated by $\hat{x}$ is equivalent to the well-known minimum cut problem, which can be nominally solved in $O(|V|^4)$ [53]. We can use this approach to implement Step 3 of the cutting plane method and hence compute the bound z_D effectively. As an example, consider the vector $\hat{x}$ pictured graphically in Figure 4, obtained in Step 2 of the cutting plane method. In the figure, only edges e for which $\hat{x}_e > 0$ are shown. Each edge e is labeled with the value $\hat{x}_e$, except for edges e with $\hat{x}_e = 1$. The circled set of vertices $S = \{0, 1, 2, 3, 7\}$ define a SEC violated by $\hat{x}$, since $\hat{x}(E(S)) = 4.6 > 4.0 = |S| - 1$. ■

3.2 Dantzig-Wolfe Method

In the Dantzig-Wolfe method, the bound z_D can be obtained by dynamically generating portions of an inner description of $\mathcal{P}'$ and intersecting it with $\mathcal{Q}''$. Consider Minkowski's Theorem, which states that every bounded polyhedron is finitely generated by its extreme points [52]. Let $\mathcal{E} \subseteq \mathcal{F}'$ be the set of extreme points of $\mathcal{P}'$, so that

$$\mathcal{P}' = \{x \in \mathbb{R}^n \mid x = \sum_{s \in \mathcal{E}} s\lambda_s, \sum_{s \in \mathcal{E}} \lambda_s = 1, \lambda_s \geq 0 \ \forall s \in \mathcal{E}\}. \tag{31}$$

Then the Dantzig-Wolfe formulation for computing the bound z_D is

$$z_{DW} = \min_{x \in \mathbb{R}^n} \{c^\top x \mid A''x \geq b'', x = \sum_{s \in \mathcal{E}} s\lambda_s, \sum_{s \in \mathcal{E}} \lambda_s = 1, \lambda_s \geq 0 \ \forall s \in \mathcal{E}\}. \tag{32}$$

By substituting out the original variables, this formulation can be rewritten in the more familiar form

$$z_{DW} = \min_{\lambda \in \mathbb{R}^{\mathcal{E}}_+} \{c^\top (\sum_{s \in \mathcal{E}} s\lambda_s) \mid A''(\sum_{s \in \mathcal{E}} s\lambda_s) \geq b'', \sum_{s \in \mathcal{E}} \lambda_s = 1\}. \tag{33}$$

This is a linear program, but since the set of extreme points $\mathcal{E}$ is potentially of exponential size, we dynamically generate those that are relevant by solving an optimization problem over $\mathcal{P}'$. An outline of the method is presented in Figure 5.

In Step 2, we solve the master problem, which is a restricted linear program obtained by substituting $\mathcal{E}^t$ for $\mathcal{E}$ in (33). In Section 6, we discuss several alternatives for solving this LP. In any case, solving it results in a primal solution λ^t_{DW}, and a dual solution consisting of the dual multipliers u^t_{DW} on the constraints corresponding to $[A'', b'']$ and the multiplier α^t_{DW} on the convexity constraint. The dual solution is needed to generate the improving columns in Step 3. In each iteration, we are generating an inner approximation, $\mathcal{P}^t_I \subseteq \mathcal{P}'$, the convex hull of $\mathcal{E}^t$. Thus $\mathcal{P}^t_I \cap \mathcal{Q}''$ may or may not contain $\mathcal{P}$ and the bound returned from the master problem in Step 2, $\bar{z}^t_{DW}$, provides an *upper* bound on z_{DW}. Nonetheless, it is easy to show (see Section 3.3) that an optimal solution to the subproblem solved in Step 3 yields a valid lower bound. In particular, if $\tilde{s}$ is a member of $\mathcal{E}$ with the smallest reduced cost in Step 3, then

$$\underline{z}^t_{DW} = c^\top \tilde{s} + (u^t_{DW})^\top (b'' - A''\tilde{s}) \tag{38}$$

is a valid lower bound. This means that, in contrast to the cutting plane method, where a valid lower bound is always available, the Dantzig-Wolfe method only yields a valid lower bound when the subproblem is solved to optimality, i.e., the optimization version is solved, as opposed to the decision version. This need not be done in every iteration, as described below.

In Step 3, we search for *improving* members of $\mathcal{E}$, where, as in the previous section, this means members that when added to $\mathcal{E}^t$ yield an improved bound. It is less clear here, however, which bound we would like to improve, $\bar{z}^t_{DW}$ or $\underline{z}^t_{DW}$. A necessary condition for improving $\bar{z}^t_{DW}$ is the generation of a column with negative reduced cost. In fact, if one considers (38), it is clear that this condition is also necessary for improvement of $\underline{z}^t_{DW}$. However, we point out again that the subproblem must be solved to optimality in order to update the bound $\underline{z}^t_{DW}$. In either case, however, we are looking for members of $\mathcal{E}$ with negative reduced cost. If one or more such members exist, we add them to $\mathcal{E}^t$ and iterate.

An area that deserves some deeper investigation is the relationship between the solution obtained by solving the reformulation (35) and the solution that would be obtained by solving an LP directly over $\mathcal{P}^t_I \cap \mathcal{Q}''$ with the objective function c. Consider the primal optimal solution λ^t_{DW}, which we refer to as an *optimal decomposition*. If we combine the members of $\mathcal{E}^t$ using λ^t_{DW} to obtain an *optimal fractional solution*

$$x^t_{DW} = \sum_{s \in \mathcal{E}^t} s(\lambda^t_{DW})_s, \tag{39}$$

then we see that $\bar{z}^t_{DW} = c^\top x^t_{DW}$. In fact, $x^t_{DW} \in \mathcal{P}^t_I \cap \mathcal{Q}''$ is an optimal solution to the linear program solved directly over $\mathcal{P}^t_I \cap \mathcal{Q}''$ with objective function c.

The optimal fractional solution plays an important role in the integrated methods to be introduced later. To illustrate the Dantzig-Wolfe method and the role of the optimal fractional solution in the method, we show how to apply it to generate the bound z_D for the ILP of Example 1.

Example 1 (Continued) For the purposes of illustration, we begin with a randomly generated initial set of points $\mathcal{E}_0 = \{(4, 1), (5, 5)\}$. Taking their convex hull, we form the

Dantzig-Wolfe Method

Input: An instance $ILP(\mathcal{P}, c)$.
Output: A lower bound z_{DW} on the optimal solution value for the instance, a primal solution $\hat{\lambda}_{DW} \in \mathbb{R}^{\mathcal{E}}$, and a dual solution $(\hat{u}_{DW}, \hat{\alpha}_{DW}) \in \mathbb{R}^{m''+1}$.

1. **Initialize**: Construct an initial inner approximation

$$\mathcal{P}_I^0 = \{\sum_{s \in \mathcal{E}^0} s\lambda_s \mid \sum_{s \in \mathcal{E}^0} \lambda_s = 1, \lambda_s \geq 0 \ \forall s \in \mathcal{E}^0, \lambda_s = 0 \ \forall s \in \mathcal{E} \setminus \mathcal{E}^0\} \subseteq \mathcal{P}' \quad (34)$$

from an initial set $\mathcal{E}^0$ of extreme points of $\mathcal{P}'$ and set $t \leftarrow 0$.

2. **Master Problem**: Solve the Dantzig-Wolfe reformulation

$$\bar{z}_{DW}^t = \min_{\lambda \in \mathbb{R}_+^{\mathcal{E}}} \{c^\top (\sum_{s \in \mathcal{E}} s\lambda_s) \mid A''(\sum_{s \in \mathcal{E}} s\lambda_s) \geq b'', \sum_{s \in \mathcal{E}} \lambda_s = 1, \lambda_s = 0 \ \forall s \in \mathcal{E} \setminus \mathcal{E}^t\} \quad (35)$$

to obtain the optimal value $\bar{z}_{DW}^t = \min_{\mathcal{P}_I^t \cap \mathcal{Q}''} c^\top x \geq z_{DW}$, an optimal primal solution $\lambda_{DW}^t \in \mathbb{R}_+^{\mathcal{E}}$, and an optimal dual solution $(u_{DW}^t, \alpha_{DW}^t) \in \mathbb{R}^{m''+1}$.

3. **Subproblem**: Call the subroutine $OPT(\mathcal{P}', c^\top - (u_{DW}^t)^\top A'', \alpha_{DW}^t)$, generating a set of $\tilde{\mathcal{E}}$ of *improving* members of $\mathcal{E}$ with negative reduced cost, where the reduced cost of $s \in \mathcal{E}$ is

$$rc(s) = (c^\top - (u_{DW}^t)^\top A'')s - \alpha_{DW}^t. \quad (36)$$

If $\tilde{s} \in \tilde{\mathcal{E}}$ is the member of $\mathcal{E}$ with smallest reduced cost, then $\underline{z}_{DW}^t = rc(\tilde{s}) + \alpha_{DW}^t + (u_{DW}^t)^\top b'' \leq z_{DW}$ provides a valid lower bound.

4. **Update**: If $\tilde{\mathcal{E}} \neq \emptyset$, set $\mathcal{E}^{t+1} \leftarrow \mathcal{E}^t \cup \tilde{\mathcal{E}}$ to form the new inner approximation

$$\mathcal{P}_I^{t+1} = \{\sum_{s \in \mathcal{E}^{t+1}} s\lambda_s \mid \sum_{s \in \mathcal{E}^{t+1}} \lambda_s = 1, \lambda_s \geq 0 \ \forall s \in \mathcal{E}^{t+1}, \lambda_s = 0 \ \forall s \in \mathcal{E} \setminus \mathcal{E}^{t+1}\} \subseteq \mathcal{P}', \quad (37)$$

and set $t \leftarrow t+1$. Go to Step 2.

5. If $\tilde{\mathcal{E}} = \emptyset$, output the bound $z_{DW} = \bar{z}_{DW}^t = \underline{z}_{DW}^t$, $\hat{\lambda}_{DW} = \lambda_{DW}^t$, and $(\hat{u}_{DW}, \hat{\alpha}_{DW}) = (u_{DW}^t, \alpha_{DW}^t)$.

Figure 5: Outline of the Dantzig-Wolfe method

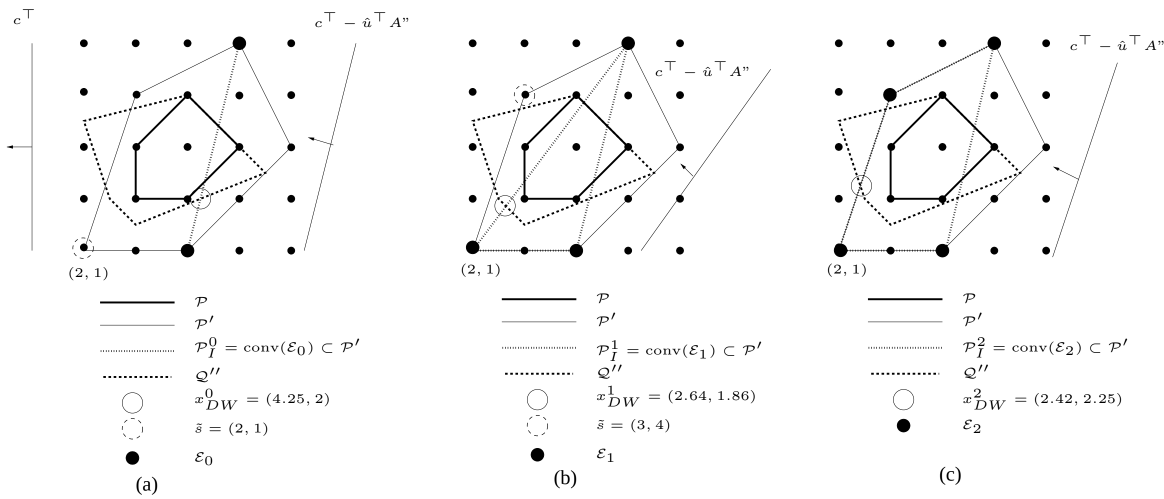


Figure 6: Dantzig-Wolfe method (Example 1)

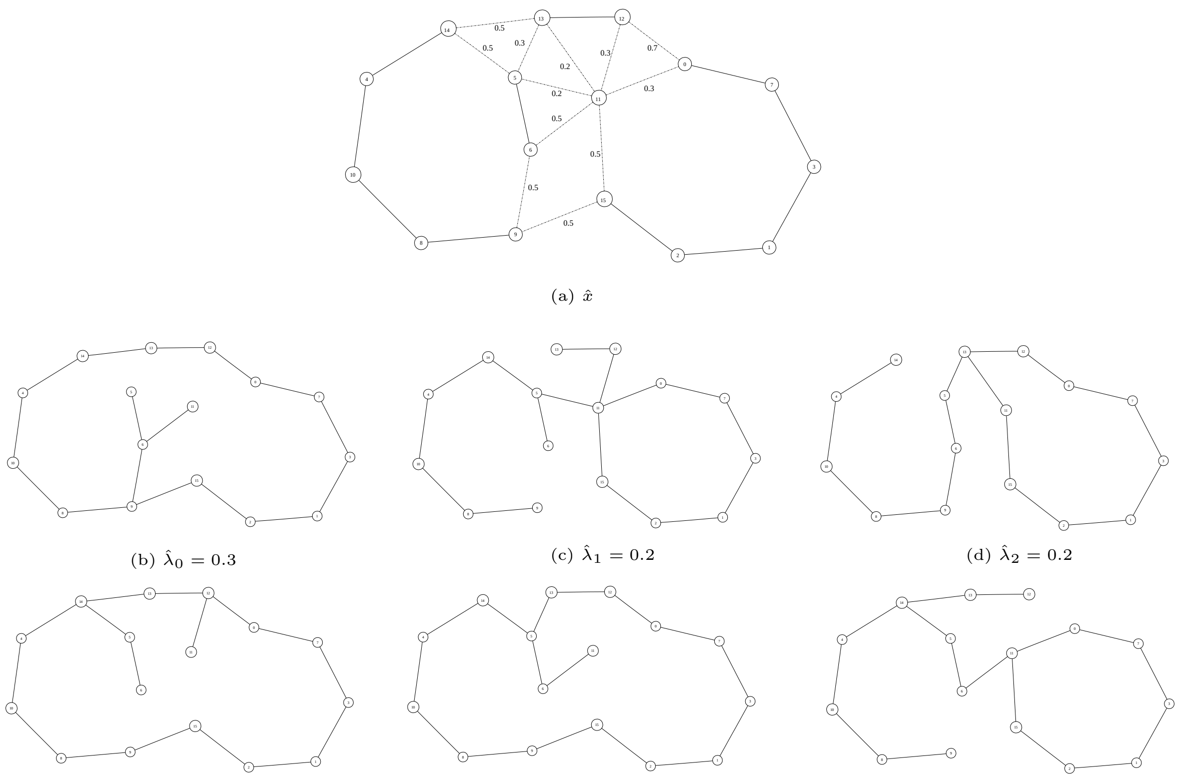
initial inner approximation $\mathcal{P}_I^0 = \mathrm{conv}(\mathcal{E}^0)$, as illustrated in Figure 6(a).

Iteration 0. Solving the master problem with inner polyhedron $\mathcal{P}_I^0$, we obtain an optimal primal solution $(\lambda_{DW}^0)_{(4,1)} = 0.75, (\lambda_{DW}^0)_{(5,5)} = 0.25$, $x_{DW}^0 = (4.25, 2)$, and bound $\bar{z}_{DW}^0 = 4.25$. Since constraint (12) is binding at x_{DW}^0, the only nonzero component of u_{DW}^0 is $(u_{DW}^0)_{(12)} = 0.28$, while the dual variable associated with the convexity constraint has value $\alpha_{DW}^0 = 4.17$. All other dual variables have value zero. Next, we search for an extreme point of $\mathcal{P}'$ with negative reduced cost, by solving the subproblem $OPT(\mathcal{P}', c^\top - (u_{DW}^t)^\top A'', \alpha_{DW}^0)$. From Figure 6(a), we see that $\tilde{s} = (2, 1)$. This gives a valid lower bound $\underline{z}_{DW}^0 = 2.03$. We add the corresponding column to the restricted master and set $\mathcal{E}^1 = \mathcal{E}^0 \cup \{(2, 1)\}$.

Iteration 1. The next iteration is depicted in Figure 6(b). First, we solve the master problem with inner polyhedron $\mathcal{P}_I^1 = \mathrm{conv}(\mathcal{E}^1)$ to obtain $(\lambda_{DW}^1)_{(5,5)} = 0.21$, $(\lambda_{DW}^1)_{(2,1)} = 0.79$, $x_{DW}^1 = (2.64, 1.86)$, and bound and $\bar{z}_{DW}^1 = 2.64$. This also provides the dual solution $(u_{DW}^1)_{(13)} = 0.43$ and $\alpha_{DW}^1 = 0.71$ (all other dual values are zero). Solving $OPT(\mathcal{P}', c^\top - u_{DW}^1 A'', \alpha_{DW}^1)$, we obtain $\tilde{s} = (3, 4)$, and $\underline{z}_{DW}^1 = 1.93$. We add the corresponding column to the restricted master and set $\mathcal{E}^2 = \mathcal{E}^1 \cup \{(3, 4)\}$.

Iteration 2 The final iteration is depicted in Figure 6(c). Solving the master problem once more with inner polyhedron $\mathcal{P}_I^2 = \mathrm{conv}(\mathcal{E}^2)$, we obtain $(\lambda_{DW}^2)_{(2,1)} = 0.58$ and $(\lambda_{DW}^2)_{(3,4)} = 0.42$, $x_{DW}^2 = (2.42, 2.25)$, and bound $\bar{z}_{DW}^2 = 2.42$. This also provides the dual solution $(u_{DW}^2)_{(14)} = 0.17$ and $\alpha_{DW}^2 = 0.83$. Solving $OPT(\mathcal{P}'.c^\top - u_{DW}^2 A'', \alpha_{DW}^2)$, we conclude that $\tilde{\mathcal{E}} = \emptyset$. We therefore terminate with the bound $z_{DW} = 2.42 = z_D$. ■

As a further brief illustration, we return to the TSP example introduced earlier.



(a) $\hat{x}$

(b) $\hat{\lambda}_0 = 0.3$ (c) $\hat{\lambda}_1 = 0.2$ (d) $\hat{\lambda}_2 = 0.2$

(e) $\hat{\lambda}_3 = 0.1$ (f) $\hat{\lambda}_4 = 0.1$ (g) $\hat{\lambda}_5 = 0.1$

Figure 7: Dantzig-Wolfe method (Example 2)

Example 2 (Continued) As we noted earlier, the Minimum 1-Tree Problem can be solved by computing a minimum cost spanning tree on vertices $V \setminus \{0\}$, and then adding two cheapest edges incident to vertex 0. This can be done in $O(|E| \log |V|)$ using standard algorithms. In applying the Dantzig-Wolfe method to compute z_D using the decomposition described earlier, the subproblem to be solved in Step 3 is a Minimum 1-Tree Problem. Because we can solve this problem effectively, we can apply the Dantzig-Wolfe method in this case. As an example of the result of solving the Dantzig-Wolfe master problem (35), Figure 7 depicts an optimal fractional solution (a) to a Dantzig-Wolfe master LP and the six extreme points 7(b-g) of the 1-tree polyhedron $\mathcal{P}'$, with nonzero weight comprising an optimal decomposition. We return to this figure later in Section 4. ∎

Now consider the set $\mathcal{S}(u, \alpha)$, defined as

$$\mathcal{S}(u, \alpha) = \{s \in \mathcal{E} \mid (c^\top - u^\top A'')s = \alpha\}, \tag{40}$$

where $u \in \mathbb{R}^{m''}$ and $\alpha \in \mathbb{R}$. The set $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$ is the set of members of $\mathcal{E}$ with reduced cost zero at optimality for (35) in iteration t. It follows that $\mathrm{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$ is in fact the face of optimal solutions to the linear program solved over $\mathcal{P}^t_I$ with objective function $c^\top - u^\top A''$. This line of reasoning culminates in the following theorem tying together the set $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$ defined above, the vector x^t_{DW}, and the optimal face of solutions to the LP over the polyhedron $\mathcal{P}^t_I \cap \mathcal{Q}''$.

Theorem 2 *conv*$(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$ *is a face of* $\mathcal{P}^t_I$ *and contains* x^t_{DW}.

Proof. We first show that $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$ is a face of $\mathcal{P}^t_I$. Observe that

$$(c^\top - (u^t_{DW})^\top A'', \alpha^t_{DW})$$

defines a valid inequality for $\mathcal{P}^t_I$ since α^t_{DW} is the optimal value for the problem of minimizing over $\mathcal{P}^t_I$ with objective function $c^\top - (u^t_{DW})^\top A''$. Thus, the set

$$G = \{x \in \mathcal{P}^t_I \mid (c^\top - (u^t_{DW})^\top A'')x = \alpha^t_{DW}\}, \tag{41}$$

is a face of $\mathcal{P}^t_I$ that contains $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$. We will show that $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW})) = G$. Since G is convex and contains $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$, it also contains $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$, so we just need to show that $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$ contains G. We do so by observing that the extreme points of G are elements of $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$. By construction, all extreme points of $\mathcal{P}^t_I$ are members of $\mathcal{E}$ and the extreme points of G are also extreme points of $\mathcal{P}^t_I$. Therefore, the extreme points of G must be members of $\mathcal{E}$ and contained in $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$. The claim follows and $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$ is a face of $\mathcal{P}^t_I$.

The fact that $x^t_{DW} \in \text{conv}(S(u^t_{DW}, \alpha^t_{DW}))$ follows from the fact that x^t_{DW} is a convex combination of members of $\mathcal{S}(u^t_{DW}, \alpha^t_{DW})$. ■

An important consequence of Theorem 2 is that the face of optimal solutions to the LP over the polyhedron $\mathcal{P}^t_I \cap \mathcal{Q}''$ is actually contained in $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW})) \cap \mathcal{Q}''$, as stated in the following corollary.

Corollary 2 *If F is the face of optimal solutions to the linear program solved directly over $\mathcal{P}^t_I \cap \mathcal{Q}''$ with objective function vector c, then $F \subseteq$ conv$(\mathcal{S}(u^t_{DW}, \alpha^t_{DW})) \cap \mathcal{Q}''$.*

Proof. Let $\hat{x} \in F$ be given. Then we have that $\hat{x} \in \mathcal{P}^t_I \cap \mathcal{Q}''$ by definition, and

$$c^\top \hat{x} = \alpha^t_{DW} + (u^t_{DW})^\top b'' = \alpha^t_{DW} + (u^t_{DW})^\top A'' \hat{x}, \tag{42}$$

where the first equality in this chain is a consequence of strong duality and the last is a consequence of complementary slackness. Hence, it follows that $(c^\top - (u^t_{DW})^\top A'')\hat{x} = \alpha^t_{DW}$ and the result is proven. ■

Hence, each iteration of the method not only produces the primal solution $x^t_{DW} \in \mathcal{P}^t_I \cap \mathcal{Q}''$, but also a dual solution $(u^t_{DW}, \alpha^t_{DW})$ that defines a face $\text{conv}(\mathcal{S}(u^t_{DW}, \alpha^t_{DW}))$ of $\mathcal{P}^t_I$ that contains the entire optimal face of solutions to the LP solved directly over $\mathcal{P}^t_I \cap \mathcal{Q}''$ with the original objective function vector c.

When no column with negative reduced cost exists, the two bounds must be equal to z_D and we stop, outputting both the primal solution $\hat{\lambda}_{DW}$, and the dual solution $(\hat{u}_{DW}, \hat{\alpha}_{DW})$. It follows from the results proven above that in the final iteration, any column of (35) with reduced cost zero must in fact have a cost of $\hat{\alpha}_{DW} = z_D - \hat{u}^\top_{DW} b''$ when evaluated with respect to the modified objective function $c^\top - \hat{u}^\top_{DW} A''$. In the final iteration, we can therefore strengthen the statement of Theorem 2, as follows.

Theorem 3 *conv*$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW}))$ *is a face of* $\mathcal{P}'$ *and contains* $\hat{x}_{DW}$*.*

The proof follows along the same lines as Theorem 2. As before, we can also state the following important corollary.

Corollary 3 *If* F *is the face of optimal solutions to the linear program solved directly over* $\mathcal{P}' \cap \mathcal{Q}''$ *with objective function vector* c*, then* $F \subseteq$ *conv*$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})) \cap \mathcal{Q}''$*.*

Thus, conv$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW}))$ is actually a face of $\mathcal{P}'$ that contains $\hat{x}_{DW}$ and the entire face of optimal solutions to the LP solved over $\mathcal{P}' \cap \mathcal{Q}''$ with objective function c. This fact provides strong intuition regarding the connection between the Dantzig-Wolfe method and the cutting plane method and allows us to regard Dantzig-Wolfe decomposition as either a procedure for producing the bound $z_D = c^\top \hat{x}_{DW}$ from primal solution information or the bound $z_D = c^\top \hat{s} + \hat{u}_{DW}^\top (b'' - A''\hat{s})$, where $\hat{s}$ is any member of $\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})$, from dual solution information. This fact is important in the next section, as well as later when we discuss integrated methods.

The exact relationship between $\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})$, the polyhedron $\mathcal{P}' \cap \mathcal{Q}''$, and the face F of optimal solutions to an LP solved over $\mathcal{P}' \cap \mathcal{Q}''$ can vary for different polyhedra and even for different objective functions. Figure 8 shows the polyhedra of Example 1 with three different objective functions indicated. The convex hull of $\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})$ is typically a proper face of $\mathcal{P}'$, but it is possible for $\hat{x}_{DW}$ to be an inner point of $\mathcal{P}'$, in which case we have the following result.

Theorem 4 *If* $\hat{x}_{DW}$ *is an inner point of* $\mathcal{P}'$*, then conv*$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})) = \mathcal{P}'$*.*

Proof. We prove the contrapositive. Suppose conv$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW}))$ 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$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})) \subseteq \{x \in \mathbb{R}^n \mid ax = \beta\}$. By Theorem 3, $\hat{x}_{DW} \in$ conv$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW}))$ and $\hat{x}_{DW}$ therefore cannot satisfy the definition of an inner point. ■

In this case, illustrated graphically in Figure 8(a) with the polyhedra from Example 1, $z_{DW} = z_{LP}$ and Dantzig-Wolfe decomposition does not improve the bound. All columns of the Dantzig-Wolfe LP have reduced cost zero and any member of $\mathcal{E}$ can be given positive weight in an optimal decomposition. 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. This condition indicates that the chosen relaxation may be too weak.

A second case of potential interest is when $F =$ conv$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})) \cap \mathcal{Q}''$, illustrated graphically in Figure 8(b). 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. This condition can be detected by examining the objective function values of the members of $\mathcal{E}$ with positive weight in the optimal decomposition. If they are all identical, any such member 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_{IP}. The more typical case, in which F is a proper subset of conv$(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})) \cap \mathcal{Q}''$, is shown in Figure 8(c).

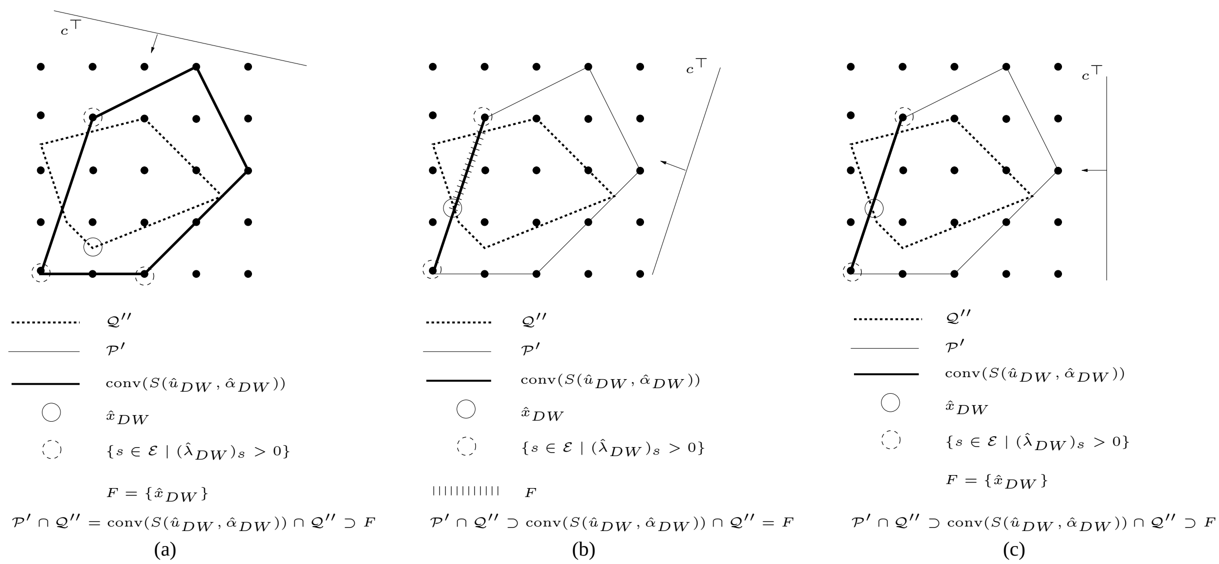


Figure 8: The relationship of $\mathcal{P}' \cap \mathcal{Q}''$, $\operatorname{conv}(\mathcal{S}(\hat{u}_{DW}, \hat{\alpha}_{DW})) \cap \mathcal{Q}''$, and the face F.

3.3 Lagrangian Method

The Lagrangian method [22, 14] is a general approach for computing z_D that is closely related to the Dantzig-Wolfe method, but is focused primarily on producing dual solution information. The Lagrangian method can be viewed as a method for producing a particular face of $\mathcal{P}'$, as in the Dantzig-Wolfe method, but no explicit approximation of $\mathcal{P}'$ is maintained. Although there are implementations of the Lagrangian method that *do* produce approximate primal solution information similar to the solution information that the Dantzig-Wolfe method produces (see Section 3.2), our viewpoint is that the main difference between the Dantzig-Wolfe method and the Lagrangian method is the type of solution information they produce. This distinction is important when we discuss integrated methods in Section 4. When exact primal solution information is not required, faster algorithms for determining the dual solution are possible. By employing a Lagrangian framework instead of a Dantzig-Wolfe framework, we can take advantage of this fact.

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

$$z_{LR}(u) = \min_{s \in \mathcal{F}'} \{c^\top s + u^\top (b'' - A'' s)\}. \tag{43}$$

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 $[A'', b'']$. Note that (43) is the same subproblem solved in the Dantzig-Wolfe method to generate the most negative reduced cost column. The problem

$$z_{LD} = \max_{u \in \mathbb{R}^{m''}_{+}} \{z_{LR}(u)\} \tag{44}$$

of maximizing this bound over all choices of dual multipliers is a dual to (1) called the

Lagrangian Method

Input: An instance $ILP(\mathcal{P}, c)$.
Output: A lower bound z_{LD} on the optimal solution value for the instance and a dual solution $\hat{u}_{LD} \in \mathbb{R}^{m''}$.

1. Let $s^0_{LD} \in \mathcal{E}$ define some initial extreme point of $\mathcal{P}'$, u^0_{LD} some initial setting for the dual multipliers and set $t \leftarrow 0$.
2. **Master Problem:** Using the solution information gained from solving the pricing subproblem, and the previous dual setting u^t_{LD}, update the dual multipliers u^{t+1}_{LD}.
3. **Subproblem:** Call the subroutine $OPT(\mathcal{P}', c^\top - (u^t_{LD})^\top A'', (c - (u^t_{LD})^\top A'')s^t_{LD})$, to solve
$$z^t_{LD} = \min_{s \in \mathcal{F}'} \{(c^\top - (u^t_{LD})^\top A'')s + b''^\top u^t_{LD}\}. \tag{45}$$
Let $s^{t+1}_{LD} \in \mathcal{E}$ be the optimal solution to this subproblem, if one is found.
4. If a prespecified stopping criterion is met, then output $z_{LD} = z^t_{LD}$ and $\hat{u}_{LD} = u^t_{LD}$, otherwise, go to Step 2

Figure 9: Outline of the Lagrangian method

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*.

It is easy to see that $z_{LR}(u)$ is a piecewise linear concave function and can be maximized by any number of methods for non-differentiable optimization. In Section 6, we discuss some alternative solution methods (for a complete treatment, see [34]). In Figure 9 we give an outline of the steps involved in the Lagrangian method. As in Dantzig-Wolfe, the main loop involves updating the dual solution and then generating an *improving* member of $\mathcal{E}$ by solving a subproblem. Unlike the Dantzig-Wolfe method, there is no approximation and hence no update step, but the method can nonetheless be viewed in the same frame of reference.

To more clearly see the connection to the Dantzig-Wolfe method, consider the dual of the Dantzig-Wolfe LP (33),

$$z_{DW} = \max_{\alpha \in \mathbb{R}, u \in \mathbb{R}^{m''}_+} \{\alpha + b''^\top u \mid \alpha \leq (c^\top - u^\top A'')s \ \forall s \in \mathcal{E}\}. \tag{46}$$

Letting $\eta = \alpha + b''^\top u$ and rewriting, we see that

$$\begin{aligned} z_{DW} &= \max_{\eta \in \mathbb{R}, u \in \mathbb{R}^{m''}_+} \{\eta \mid \eta \leq (c^\top - u^\top A'')s + b''^\top u \ \forall s \in \mathcal{E}\} & (47)\\ &= \max_{\eta \in \mathbb{R}, u \in \mathbb{R}^{m''}_+} \{\min_{s \in \mathcal{E}} \{(c^\top - u^\top A'')s + b''^\top u\}\} = z_{LD}. & (48)\end{aligned}$$

Thus, we have that $z_{LD} = z_{DW}$ and that (44) is another formulation for the problem of calculating z_D. It is also interesting to observe that the set $\mathcal{S}(u^t_{LD}, z^t_{LD} - b''^\top u^t_{LD})$ is the set of alternative optimal solutions to the subproblem solved at iteration t in Step 3. The following theorem is a counterpart to Theorem 3 that follows from this observation.

Theorem 5 *$conv(\mathcal{S}(\hat{u}_{LD}, z_{LD} - b''^\top \hat{u}_{LD}))$ is a face of $\mathcal{P}'$. Also, if F is the face of optimal solutions to the linear program solved directly over $\mathcal{P}' \cap \mathcal{Q}''$ with objective function vector c, then $F \subseteq conv(\mathcal{S}(\hat{u}_{LD}, z_{LD} - b''^\top \hat{u}_{LD})) \cap \mathcal{Q}''$.*

Again, the proof is similar to that of Theorem 3. This shows that while the Lagrangian method does not maintain an explicit approximation, it does produce a face of $\mathcal{P}'$ containing the optimal face of solutions to the linear program solved over the approximation $\mathcal{P}' \cap \mathcal{Q}''$.

4 Integrated Decomposition Methods

In Section 3, we demonstrated that traditional decomposition approaches can be viewed as utilizing dynamically generated polyhedral information to improve the LP bound by either building an inner or an outer approximation of an implicitly defined polyhedron that approximates $\mathcal{P}$. The choice between inner and outer methods is largely an empirical one, but recent computational research has favored outer methods. In what follows, we discuss three methods for integrating inner and outer methods. In principle, this is not difficult to do and can result in bounds that are improved over those achieved by either approach alone.

While traditional decomposition approaches build either an inner *or* an outer approximation, *integrated decomposition methods* build both an inner *and* an outer approximation. These methods follow the same basic loop as traditional decomposition methods, except that the master problem is required to generate both primal *and* dual solution information and the subproblem can be either a separation problem *or* an optimization problem. The first two techniques we describe integrate the cutting plane method with either the Dantzig-Wolfe method or the Lagrangian method. The third technique, described in Section 5, is a cutting plane method that uses an inner approximation to perform separation.

4.1 Price and Cut

The integration of the cutting plane method with the Dantzig-Wolfe method results in a procedure that alternates between a subproblem that generates improving columns (the *pricing* subproblem) and a subproblem that generates improving valid inequalities (the *cutting* subproblem). Hence, 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 already been studied previously by a number of authors [12, 61, 38, 11, 60] and more recently by Arãgao and Uchoa [21].

As in the Dantzig-Wolfe method, the bound produced by price and cut can be thought of as resulting from the intersection of two approximating polyhedra. However, the Dantzig-Wolfe method required one of these, $\mathcal{Q}''$, to have a short description. With integrated methods, both polyhedra can have descriptions of exponential size. Hence, price and cut allows partial descriptions of both an inner polyhedron $\mathcal{P}_I$ *and* an outer polyhedron $\mathcal{P}_O$

to be generated dynamically. To optimize over the intersection of $\mathcal{P}_I$ and $\mathcal{P}_O$, we use a Dantzig-Wolfe reformulation as in (33), except that the $[A'', b'']$ is replaced by a matrix that changes dynamically. The outline of this method is shown in Figure 10.

In examining the steps of this generalized method, the most interesting question that arises is how methods for generating improving columns and valid inequalities translate to this new dynamic setting. Potentially troublesome is the fact that column generation results in a reduction of the bound $\bar{z}^t_{PC}$ produced by (51), while generation of valid inequalities is aimed at increasing it. Recall again, however, that while it is the bound $\bar{z}^t_{PC}$ that is directly produced by solving (51), it is the bound $\underline{z}^t_{PC}$ obtained by solving the pricing subproblem that one might claim is more relevant to our goal and this bound can be potentially improved by generation of either valid inequalities or columns.

Improving columns can be generated in much the same way as they were in the Dantzig-Wolfe method. To search for new columns, we simply look for those with negative reduced cost, where reduced cost is defined to be the usual LP reduced cost with respect to the current reformulation. Having a negative reduced cost is still a necessary condition for a column to be improving. However, it is less clear how to generate improving valid inequalities. Consider an optimal fractional solution x^t_{PC} obtained by combining the members of $\mathcal{E}$ according to weights yielded by the optimal decomposition λ^t_{PC} in iteration t. Following a line of reasoning similar to that followed in analyzing the results of the Dantzig-Wolfe method, we can conclude that x^t_{PC} is in fact an optimal solution to an LP solved directly over $\mathcal{P}^t_I \cap \mathcal{P}^t_O$ with objective function vector c and that therefore, it follows from Theorem 1 that any improving inequality must be violated by x^t_{PC}. It thus seems sensible to consider separating x^t_{PC} from $\mathcal{P}$. This is the approach taken in the method of Figure 10.

To demonstrate how the price and cut method works, we return to Example 1.

Example 1 (Continued) We pick up the example at the last iteration of the Dantzig-Wolfe method and show how the bound can be further improved by dynamically generating valid inequalities.

Iteration 0. Solving the master problem with $\mathcal{E}^0 = \{(4,1), (5,5), (2,1), (3,4)\}$ and the initial inner approximation $\mathcal{P}^0_I = \text{conv}(\mathcal{E}^0)$ yields $(\lambda^0_{PC})_{(2,1)} = 0.58$ and $(\lambda^0_{PC})_{(3,4)} = 0.42$, $x^0_{PC} = (2.42, 2.25)$, bound $\underline{z}^0_{PC} = \bar{z}^0_{PC} = 2.42$. Next, we solve the cutting subproblem $SEP(\mathcal{P}, x^0_{PC})$, generating facet-defining inequalities of $\mathcal{P}$ that are violated by x^0_{PC}. One such facet-defining inequality, $x_1 \geq 3$, is illustrated in Figure 11(a). We add this inequality to the current set $D^0 = [A'', b'']$ to form a new outer approximation $\mathcal{P}^1_O$, defined by the set D^1.

Iteration 1. Solving the new master problem, we obtain an optimal primal solution $(\lambda^1_{PC})_{(4,1)} = 0.42$, $(\lambda^1_{PC})_{(2,1)} = 0.42$, $(\lambda^1_{PC})_{(3,4)} = 0.17$, $x^1_{PC} = (3, 1.5)$, bound $\bar{z}^1_{PC} = 3$, as well as an optimal dual solution $(u^1_{PC}, \alpha^1_{PC})$. Next, we consider the pricing subproblem. Since x^1_{PC} is in the interior of $\mathcal{P}'$, every extreme point of $\mathcal{P}'$ has reduced cost 0 by Theorem 4. Therefore, there are no negative reduced cost columns and we switch again to the cutting subproblem $SEP(\mathcal{P}, x^1_{PC})$. As illustrated in Figure 11(b), we find another facet-defining inequality of $\mathcal{P}$ violated by x^1_{PC}, $x_2 \geq 2$. We then add this inequality to form D^2 and further tighten the outer approximation, now $\mathcal{P}^2_O$.

Price and Cut Method

Input: An instance $ILP(\mathcal{P}, c)$.
Output: A lower bound z_{PC} on the optimal solution value for the instance, a primal solution $\hat{x}_{PC} \in \mathbb{R}^n$, an optimal decomposition $\hat{\lambda}_{PC} \in \mathbb{R}^{\mathcal{E}}$, a dual solution $(\hat{u}_{PC}, \hat{\alpha}_{PC}) \in \mathbb{R}^{m^t+1}$, and the inequalities $[D_{PC}, d_{PC}] \in \mathbb{R}^{m^t \times (n+1)}$.

1. **Initialize**: Construct an initial inner approximation

$$\mathcal{P}_I^0 = \{\sum_{s \in \mathcal{E}^0} s\lambda_s \mid \sum_{s \in \mathcal{E}^0} \lambda_s = 1, \lambda_s \geq 0 \ \forall s \in \mathcal{E}^0, \lambda_s = 0 \ \forall s \in \mathcal{E} \setminus \mathcal{E}^0\} \subseteq \mathcal{P}' \quad (49)$$

from an initial set $\mathcal{E}^0$ of extreme points of $\mathcal{P}'$ *and* an initial outer approximation

$$\mathcal{P}_O^0 = \{x \in \mathbb{R}^n \mid D^0 x \geq d^0\} \supseteq \mathcal{P}, \quad (50)$$

where $D^0 = A''$ and $d^0 = b''$, and set $t \leftarrow 0$, $m^0 = m''$.

2. **Master Problem**: Solve the Dantzig-Wolfe reformulation

$$\bar{z}_{PC}^t = \min_{\lambda \in \mathbb{R}_+^{\mathcal{E}}} \{c^\top (\sum_{s \in \mathcal{E}} s\lambda_s) \mid D^t(\sum_{s \in \mathcal{E}} s\lambda_s) \geq d^t, \sum_{s \in \mathcal{E}} \lambda_s = 1, \lambda_s = 0 \ \forall s \in \mathcal{E} \setminus \mathcal{E}^t\} \quad (51)$$

of the LP over the polyhedron $\mathcal{P}_I^t \cap \mathcal{P}_O^t$ to obtain the optimal value $\bar{z}_{PC}^t$, an optimal primal solution $\lambda_{PC}^t \in \mathbb{R}^{\mathcal{E}}$, an optimal fractional solution $x_{PC}^t = \sum_{s \in \mathcal{E}} s(\lambda_{PC}^t)_s$, and an optimal dual solution $(u_{PC}^t, \alpha_{PC}^t) \in \mathbb{R}^{m^t+1}$.

3. Do either (a) or (b).

   (a) **Pricing Subproblem and Update**: Call the subroutine $OPT(\mathcal{P}', c^\top - (u_{PC}^t)^\top D^t, \alpha_{PC}^t)$, generating a set $\tilde{\mathcal{E}}$ of *improving* members of $\mathcal{E}$ with negative reduced cost (defined in Figure 5). If $\tilde{\mathcal{E}} \neq \emptyset$, set $\mathcal{E}^{t+1} \leftarrow \mathcal{E}^t \cup \tilde{\mathcal{E}}$ to form a new inner approximation $\mathcal{P}_I^{t+1}$. If $\tilde{s} \in \mathcal{E}$ is the member of $\mathcal{E}$ with smallest reduced cost, then $\underline{z}_{PC}^t = rc(\tilde{s}) + \alpha_{PC}^t + (d^t)^\top u_{PC}^t$ provides a valid lower bound. Set $[D^{t+1}, d^{t+1}] \leftarrow [D^t, d^t]$, $\mathcal{P}_O^{t+1} \leftarrow \mathcal{P}_O^t$, $m^{t+1} \leftarrow m^t$, $t \leftarrow t+1$. and go to Step 2.

   (b) **Cutting Subproblem and Update**: Call the subroutine $SEP(\mathcal{P}, x_{PC}^t)$ to generate a set of *improving* valid inequalities $[\tilde{D}, \tilde{d}] \in \mathbb{R}^{\tilde{m} \times n+1}$ for $\mathcal{P}$, violated by x_{PC}^t. If violated inequalities were found, set $[D^{t+1}, d^{t+1}] \leftarrow \left[\begin{smallmatrix} D^t & d^t \\ \tilde{D} & \tilde{d} \end{smallmatrix}\right]$ to form a new outer approximation $\mathcal{P}_O^{t+1}$. Set $m^{t+1} \leftarrow m^t + \tilde{m}$, $\mathcal{E}^{t+1} \leftarrow \mathcal{E}^t$, $\mathcal{P}_I^{t+1} \leftarrow \mathcal{P}_I^t$, $t \leftarrow t+1$, and go to Step 2.

4. If $\tilde{\mathcal{E}} = \emptyset$ and no valid inequalities were found, output the bound $z_{PC} = \bar{z}_{PC}^t = \underline{z}_{PC}^t = c^\top x_{PC}^t$, $\hat{x}_{PC} = x_{PC}^t$, $\hat{\lambda}_{PC} = \lambda_{PC}^t$, $(\hat{u}_{PC}, \hat{\alpha}_{PC}) = (u_{PC}^t, \alpha_{PC}^t)$, and $[D_{PC}, d_{PC}] = [D^t, d^t]$.

Figure 10: Outline of the price and cut method

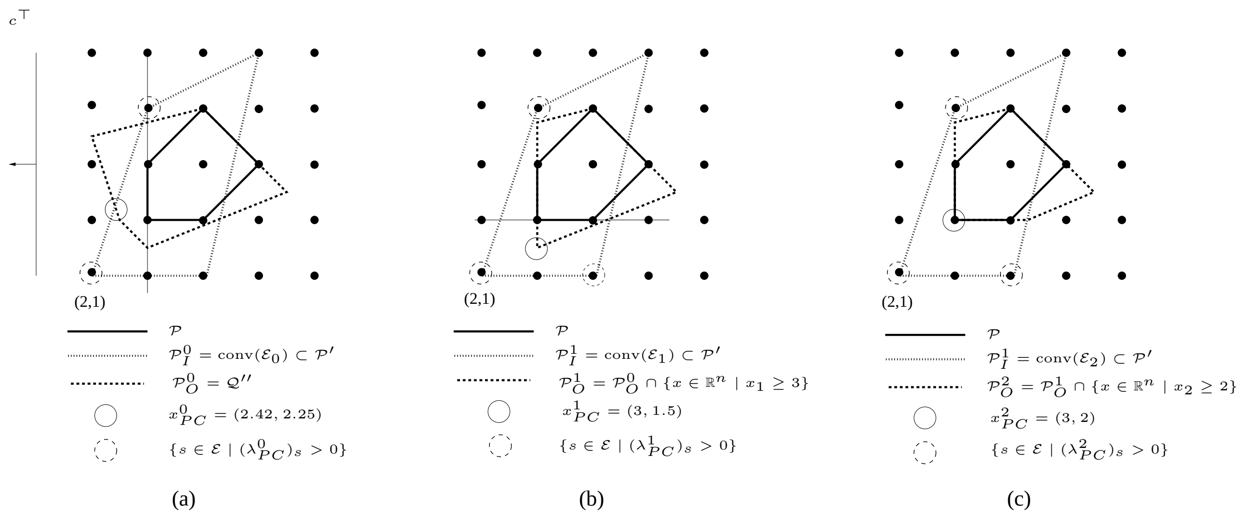


Figure 11: Price and cut method (Example 1)

Iteration 2. In the final iteration, we solve the master problem again to obtain $(\lambda^2_{PC})_{(4,1)} = 0.33$, $(\lambda^2_{PC})_{(2,1)} = 0.33$, $(\lambda^2_{PC})_{(3,4)} = 0.33$, $x^2_{PC} = (3, 2)$, bound $\bar{z}^2_{PC} = 3$. Now, since the primal solution is integral, and is contained in $\mathcal{P}' \cap \mathcal{Q}''$, we know that $z_{PC} = \bar{z}^2_{PC} = z_{IP}$ and we terminate. ■

Let us now return to the TSP example to further explore the use of the price and cut method.

Example 2 (Continued) As described earlier, application of the Dantzig-Wolfe method along with the 1-tree relaxation for the TSP allows us to compute the bound z_D obtained by optimizing over the intersection of the 1-tree polyhedron (the inner polyhedron) with the polyhedron $\mathcal{Q}''$ (the outer polyhedron) defined by constraints (17) and (19). With price and cut, we can further improve the bound by allowing both the inner and outer polyhedra to have large descriptions. For this purpose, let us now introduce the well-known *comb inequalities* [30, 31], which we will generate to improve our outer approximation. A comb is defined by a set $H \subset V$, called the *handle* and sets $T_1, T_2, ..., T_k \subset V$, called the *teeth*, which satisfy

$$\begin{aligned} H \cap T_i \neq \emptyset \ &\text{ for } i = 1, ..., k, \\ T_i \setminus H \neq \emptyset \ &\text{ for } i = 1, ..., k, \\ T_i \cap T_j = \emptyset \ &\text{ for } 1 \leq i < j \leq k, \end{aligned}$$

for some odd $k \geq 3$. Then, for $|V| \geq 6$ the comb inequality,

$$x(E(H)) + \sum_{i=1}^{k} x(E(T_i)) \leq |H| + \sum_{i=1}^{k} (|T_i| - 1) - \lceil k/2 \rceil \tag{52}$$

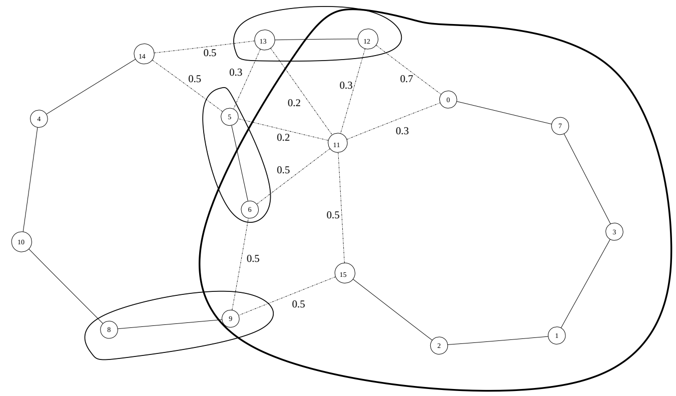


Figure 12: Price and cut method (Example 2)

is valid and facet-defining for the TSP. Let the comb polyhedron be defined by constraints (17), (19), and (52).

There are no known efficient algorithms for solving the general facet identification problem for the comb polyhedron. To overcome this difficulty, one approach is to focus on comb inequalities with special forms. One subset of the comb inequalities, known as the *blossom inequalities*, is obtained by restricting the teeth to have exactly two members. The facet identification for the polyhedron comprised of the blossom inequalities and constraints (17) and (19) can be solved in polynomial time, a fact we return to shortly. Another approach is to use heuristic algorithms not guaranteed to find a violated comb inequality when one exists (see [4] for a survey). These heuristic algorithms could be applied in price and cut as part of the cutting subproblem in Step 3b to improve the outer approximation.

In Figure 7 of Section 3.2, we showed an optimal fractional solution $\hat{x}$ that resulted from the solution of a Dantzig-Wolfe master problem and the corresponding optimal decomposition, consisting of six 1-trees. In Figure 12, we show the sets $H = \{0, 1, 2, 3, 6, 7, 9, 11, 12, 15\}, T_1 = \{5, 6\}, T_2 = \{8, 9\}$, and $T_3 = \{12, 13\}$ forming a comb that is violated by this fractional solution, since

$$\hat{x}(E(H)) + \sum_{i=1}^{k} \hat{x}(E(T_i)) = 11.3 > 11 = |H| + \sum_{i=1}^{k}(|T_i| - 1) - \lceil k/2 \rceil.$$

Such a violated comb inequality, if found, could be added to the description of the outer polyhedron to improve on the bound z_D. This shows the additional power of price and cut over the Dantzig-Wolfe method. Of course, it should be noted that it is also possible to generate such inequalities in the standard cutting plane method and to achieve the same bound improvement. ■

The choice of relaxation has a great deal of effect on the empirical behavior of decomposition algorithms. In Example 2, we employed an inner polyhedron with integer extreme points. With such a polyhedron, the integrality constraints of the inner polyhedron have no effect and $z_D = z_{LP}$. In Example 3, we consider a relaxation for which the bound z_D may be strictly improved over z_{LP} by employing an inner polyhedron that is not integral.

Example 3 Let G be a graph as defined in Example 2 for the TSP. A *2-matching* is a subgraph in which every vertex has degree two. Every TSP tour is hence a 2-matching. The *Minimum 2-Matching Problem* is a relaxation of TSP whose feasible region is described by the degree (17), bound (19), and integrality constraints (20) of the TSP. Interestingly, the 2-matching polyhedron, which is implicitly defined to be the convex hull of the feasible region just described, can also be described by replacing the integrality constraints (20) with the blossom inequalities. Just as the SEC constraints provide a complete description of the 1-tree polyhedron, the blossom inequalities (plus degree and bound) constraints provide a complete description of the 2-matching polyhedron. Therefore, we could use this polyhedron as an outer approximation to the TSP polyhedron. In [50], Müller-Hannemann and Schwartz present several polynomial algorithms for optimizing over the 2-matching polyhedron. We can therefore also use the 2-matching relaxation in the context of price and cut to generate an inner approximation of the TSP polyhedron. Using integrated methods, it would then be possible to simultaneously build up an outer approximation of the TSP polyhedron consisting of the SECs (18). Note that this simply reverses the roles of the two polyhedra from Example 2 and thus would yield the same bound.

Figure 13 shows an optimal fractional solution arising from the solution of the master problem and the 2-matchings with positive weight in a corresponding optimal decomposition. Given this fractional subgraph, we could employ the separation algorithm discussed in Example 2 of Section 3.1 to generate the violated subtour $S = \{0, 1, 2, 3, 7\}$. ■

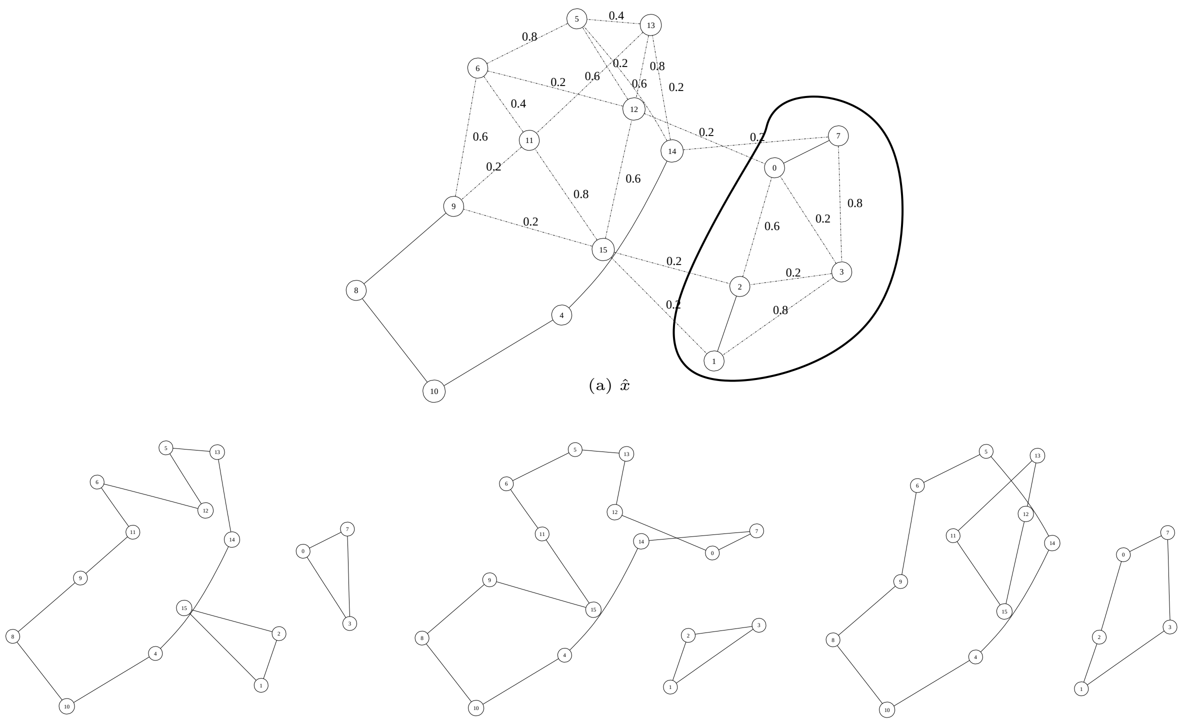
Another approach to generating improving inequalities in price and cut is to try to take advantage of the information contained in the optimal decomposition to aid in the separation procedure. This information, though computed by solving (51) is typically ignored. Consider the fractional solution x^t_{PC} generated in iteration t of the method in Figure 10. The optimal decomposition for the master problem in iteration t, λ^t_{PC}, provides a decomposition of x^t_{PC} into a convex combination of members of $\mathcal{E}$. We refer to elements of $\mathcal{E}$ that have a positive weight in this combination as *members of the decomposition*. The following theorem shows how such a decomposition can be used to derive an alternate necessary condition for an inequality to be improving. Because we apply this theorem in a more general context later in the paper, we state it in a general form.

Theorem 6 *If $\hat{x} \in \mathbb{R}^n$ violates the inequality $(a, \beta) \in \mathbb{R}^{(n+1)}$ and $\hat{\lambda} \in \mathbb{R}^{\mathcal{E}}_+$ is such that $\sum_{s\in\mathcal{E}} \hat{\lambda}_s = 1$ and $\hat{x} = \sum_{s\in\mathcal{E}} s\hat{\lambda}_s$, then there must exist an $s \in \mathcal{E}$ with $\hat{\lambda}_s > 0$ such that s also violates the inequality (a, β) .*

Proof. Let $\hat{x} \in \mathbb{R}^n$ and $(a, \beta) \in \mathbb{R}^{(n+1)}$ be given such that $a^\top \hat{x} < \beta$. Also, let $\hat{\lambda} \in \mathbb{R}^{\mathcal{E}}_+$ be given such that $\sum_{s\in\mathcal{E}} \hat{\lambda}_s = 1$ and $\hat{x} = \sum_{s\in\mathcal{E}} s\hat{\lambda}_s$. Suppose that $a^\top s \geq \beta$ for all $s \in \mathcal{E}$ with $\hat{\lambda}_s > 0$. Since $\sum_{s\in\mathcal{E}} \hat{\lambda}_s = 1$, we have $a^\top(\sum_{s\in\mathcal{E}} s\hat{\lambda}_s) \geq \beta$. Hence, $a^\top \hat{x} = a^\top(\sum_{s\in\mathcal{E}} s\hat{\lambda}_s) \geq \beta$, which is a contradiction. ■

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 in iteration t, then the following corollary is a direct consequence of Theorem 6.

Corollary 4 $\mathcal{I} \subseteq \mathcal{V} = \{(a, \beta) \in \mathbb{R}^{(n+1)} : a^\top s < \beta$ *for some* $s \in \mathcal{E}$ *such that* $(\lambda^t_{PC})_s > 0\}$.



(a) $\hat{x}$

(b) $\hat{\lambda}_0 = 0.2$

(c) $\hat{\lambda}_1 = 0.2$

(d) $\hat{\lambda}_2 = 0.6$

Figure 13: Finding violated inequalities in price and cut (Example 3)

The importance of these results is that in many cases, it is easier to separate members of $\mathcal{F}'$ from $\mathcal{P}$ than to separate arbitrary real vectors. There are a number of well-known polyhedra for which the problem of separating an arbitrary real vector is difficult, but the problem of separating a solution to a given relaxation is easy. This concept is formalized in Section 5 and some examples are discussed in Section 8. In Figure 14, we propose a new separation procedure that can be embedded in price and cut that takes advantage of this fact. The procedure takes as input an arbitrary real vector $\hat{x}$ that has been previously decomposed into a convex combination of vectors with known structure. In price and cut, the arbitrary real vector is x^t_{PC} and it is decomposed into a convex combination of members of $\mathcal{E}$ by solving the master problem (51). Rather than separating x^t_{PC} directly, the procedure consists of separating each one of the members of the decomposition in turn, then checking each inequality found for violation against x^t_{PC}.

The running time of this procedure depends in part on the cardinality of the decomposition. Carathéodory's Theorem assures us that there exists a decomposition with less than or equal to $\dim(\mathcal{P}^t_I) + 1$ members. Unfortunately, even if we limit our search to a particular known class of valid inequalities, the number of such inequalities violated by each member of $\mathcal{D}$ in Step 2 may be extremely large and these inequalities may not be violated by x^t_{PC} (such an inequality cannot be improving). Unless we enumerate *every* inequality in the set $\mathcal{V}$ from Corollary 4, either implicitly or explicitly, the procedure does not guarantee that an improving inequality will be found, even if one exists. In cases where it is possible to

Separation using a Decomposition

Input: A decomposition $\lambda \in \mathbb{R}^{\mathcal{E}}$ of $\hat{x} \in \mathbb{R}^n$.
Output: A set $[D, d]$ of potentially improving inequalities.

1. Form the set $\mathcal{D} = \{s \in \mathcal{E} \mid \lambda_s > 0\}$.
2. For each $s \in \mathcal{D}$, call the subroutine $SEP(\mathcal{P}, s)$ to obtain a set $[\tilde{D}, \tilde{d}]$ 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 14: Solving the cutting subproblem with the aid of a decomposition

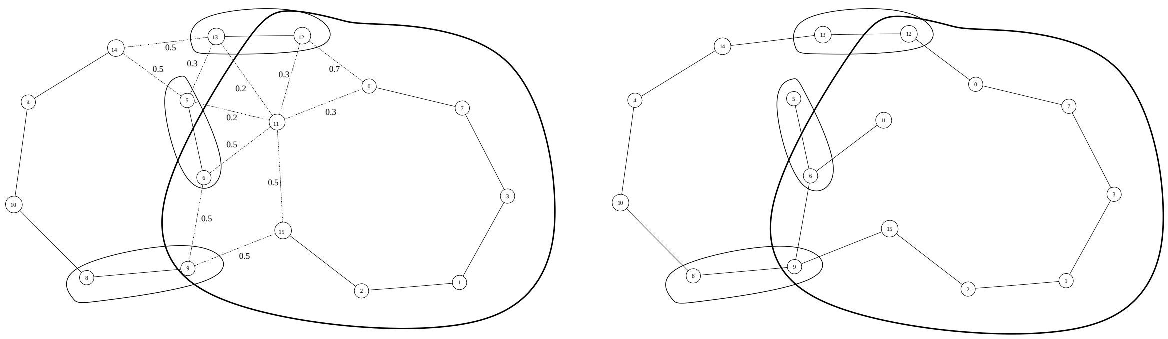
examine the set $\mathcal{V}$ in polynomial time, the worst-case complexity of the entire procedure is polynomially equivalent to that of optimizing over $\mathcal{P}'$. Obviously, it is unlikely that the set $\mathcal{V}$ can be examined in polynomial time in situations when separating x^t_{PC} is itself an $\mathcal{NP}$-complete problem. In such cases, the procedure to select inequalities that are likely to be violated by x^t_{PC} in Step 2 is necessarily a problem-dependent heuristic. The effectiveness of such heuristics can be improved in a number of ways, some of which are discussed in [57].

Note that members of the decomposition in iteration t must belong to the set $\mathcal{S}(u^t_{PC}, \alpha^t_{PC})$, as defined by (40). It follows that the convex hull of the decomposition is a subset of $\text{conv}(\mathcal{S}(u^t_{PC}, \alpha^t_{PC}))$ that contains x^t_{PC} and can be thought of as a surrogate for the face of optimal solutions to an LP solved directly over $\mathcal{P}^t_I \cap \mathcal{P}^t_O$ with objective function vector c. Combining this corollary with Theorem 1, we conclude that separation of $\mathcal{S}(u^t_{PC}, \alpha^t_{PC})$ 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 14.

Example 1 (Continued) Returning to the cutting subproblem in iteration 0 of the price and cut method, we have a decomposition $x^0_{PC} = (2.42, 2.25) = 0.58(2,1) + 0.42(3,4)$, as depicted in Figure 11(a). Now, instead of trying to solve the subproblem $SEP(\mathcal{P}, x^0_{PC})$, we instead solve $SEP(\mathcal{P}, s)$, for each $s \in \mathcal{D} = \{(2,1), (3,4)\}$. In this case, when solving the separation problem for $s = (2,1)$, we find the same facet-defining inequality of $\mathcal{P}$ as we did by separating x^0_{PC} directly.

Similarly, in iteration 1, we have a decomposition of $x^2_{PC} = (3, 1.5)$ into a convex combination of $\mathcal{D} = \{(4,1), (2,1), (3,4)\}$. Clearly, solving the separation problem for either $(2,1)$ or $(4,1)$ produces the same facet-defining inequality as with the original method. ■

Example 2 (Continued) Returning again to Example 2, recall the optimal fractional solution and the corresponding optimal decomposition arising during solution of the TSP



(a) $\hat{x}$ (a) $\hat{\lambda}_0$

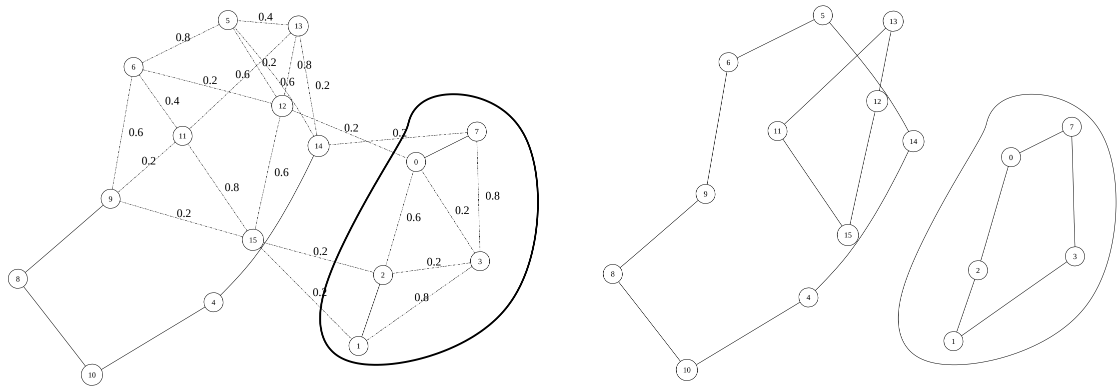
Figure 15: Using the optimal decomposition to find violated inequalities in price and cut (Example 2)

by the Dantzig-Wolfe method in Figure 7. Figure 12 shows a comb inequality violated by this fractional solution. By Theorem 6, at least one of the members of the optimal decomposition shown in Figure 7 must also violate this inequality. In fact, the member with index 0, also shown in Figure 15, is the only such member. Note that the violation is easy to discern from the structure of this integral solution. Let $\hat{x} \in \{0,1\}^E$ be the incidence vector of a 1-tree. Consider a subset H of V whose induced subgraph in the 1-tree is a path with edge set P. Consider also an odd set O of edges of the 1-tree of cardinality at least 3 and disjoint from P, such that each edge has one endpoint in H and one endpoint in $V \setminus H$. Taking the set H to be the handle and the endpoints of each member of O to be the teeth, it is easy to verify that the corresponding comb inequality will be violated by the 1-tree, since

$$\hat{x}(E(H)) + \sum_{i=1}^{k} \hat{x}(E(T_i)) = |H| - 1 + \sum_{i=1}^{k}(|T_i| - 1) > |H| + \sum_{i=1}^{k}(|T_i| - 1) - \lceil k/2 \rceil.$$

Hence, searching for such configurations in the members of the decomposition, as suggested in the procedure of Figure 14, may lead to the discovery of comb inequalities violated by the optimal fractional solution. In this case, such a configuration does in fact lead to discovery of the previously indicated comb inequality. Note that we have restricted ourselves in the above discussion to the generation of blossom inequalities. The teeth, as well as the handles can have more general forms that may lead to the discovery of more general forms of violated combs. ■

Example 3 (Continued) Returning now to Example 3, recall the optimal fractional solution and the corresponding optimal decomposition, consisting of the 2-matchings shown in Figure 13. Previously, we produced a set of vertices defining a SEC violated by the fractional point by using a minimum cut algorithm with the optimal fractional solution as input. Now, let us consider applying the procedure of Figure 14 by examining the members of the decomposition in order to discovered inequalities violated by the optimal fractional solution. Let $\hat{x} \in \{0,1\}^E$ be the incidence vector of a 2-matching. If the corresponding



(a) $\hat{x}$ (b) $\hat{\lambda}_2$

Figure 16: Using the optimal decomposition to find violated inequalities in price and cut (Example 3)

subgraph does not form a tour, then it must be disconnected. The vertices corresponding to any connected component thus define a violated SEC. By determining the connected components of each member of the decomposition, it is easy to find violated SECs. In fact, for any 2-matching, every component of the 2-matching forms a SEC that is violated by exactly 1. For the 2-matching corresponding to $\hat{s}$, we have $\hat{x}(E(S)) = |S| > |S| - 1$. Figure 16(b) shows the third member of the decomposition along with a violated SEC defined by one of its components. This same SEC is also violated by the optimal fractional solution. ■

There are many variants of the price and cut method shown in Figure 10. Most significant is the choice of which subproblem to execute during Step 3. It is easy to envision a number of heuristic rules for deciding this. For example, one obvious rule is to continue generating columns until no more are available and then switch to valid inequalities for one iteration, then generate columns again until none are available. This can be seen as performing a "complete" dual solution update before generating valid inequalities. Further variants can be obtained by not insisting on a "complete" dual update before solving the pricing problem [29, 17]. This rule could easily be inverted to generate valid inequalities until no more are available and then generate columns. A hybrid rule in which some sort of alternation occurs is a third option. The choice between these options is primarily empirical.

4.2 Relax and Cut

Just as with the Dantzig-Wolfe method, the Lagrangian method of Figure 9 can be integrated with the cutting plane method to yield a procedure several authors have termed *relax and cut*. This is done in much the same fashion as in price and cut, with a choice in each iteration between solving a pricing subproblem and a cutting subproblem. In each iteration that the cutting subproblem is solved, the generated valid inequalities are added to the description of the outer polyhedron, which is explicitly maintained as the algorithm proceeds.

As with the traditional Lagrangian method, no explicit inner polyhedron is maintained, but the algorithm can again be seen as one that computes a face of the implicitly defined inner polyhedron that contains the optimal face of solutions to a linear program solved over the intersection of the two polyhedra. When employed within a branch and bound framework, we call the overall method *branch, relax, and cut*.

An outline of the relax and cut method is shown in Figure 17. The question again arises as to how to ensure that the inequalities being generated in the cutting subproblem are improving. In the case of the Lagrangian method, this is a much more difficult issue since we cannot assume the availability of the same primal solution information available within price and cut. Furthermore, we cannot verify the condition of Corollary 1, which is the best available necessary condition for an inequality to be improving. Nevertheless, *some* primal solution information is always available in the form of the solution s^t_{RC} to the last pricing subproblem that was solved. Intuitively, separating s^t_{RC} makes sense since the infeasibilities present in s^t_{RC} may possibly be removed through the addition of valid inequalities violated by s^t_{RC}.

As with both the cutting plane and price and cut methods, the difficulty is that the valid inequalities generated by separating s^t_{RC} from $\mathcal{P}$ may not be improving, as Guignard first observed in [33]. To deepen understanding of the potential effectiveness of the valid inequalities generated, we further examine the relationship between s^t_{RC} and x^t_{PC} by recalling again the results from Section 3.2. Consider the set $\mathcal{S}(u^t_{RC}, z^t_{RC})$, where z^t_{RC} is obtained by solving the pricing subproblem (54) from Figure 17 and the set $\mathcal{S}(\cdot,\cdot)$ is as defined in (40). In each iteration where the pricing subproblem is solved, s^{t+1}_{RC} is a member of $\mathcal{S}(u^t_{RC}, z^t_{RC})$. In fact, $\mathcal{S}(u^t_{RC}, z^t_{RC})$ is exactly the set of alternative solutions to this pricing subproblem. In price and cut, a number of members of this set are available, one of which must be violated in order for a given inequality to be improving. This yields a verifiable necessary condition for a generated inequality to be improving. Relax and cut, in its most straightforward incarnation, produces one member of this set. Even if improving inequalities exist, it is possible that none of them are violated by the member of $\mathcal{S}(u^t_{RC}, z^t_{RC})$ so produced, especially if it would have had a small weight in the optimal decomposition produced by the corresponding iteration of price and cut.

It is important to note that by keeping track of the solutions to the Lagrangian subproblem that are produced while solving the Lagrangian dual, one can approximate the optimal decomposition and the optimal fractional solution produced by solving (51). This is the approach taken by the volume algorithm [9] and a number of other subgradient-based methods. As in price and cut, when this fractional solution is an inner point of $\mathcal{P}'$, all members of $\mathcal{F}'$ are alternative optimal solutions to the pricing subproblem and the bound is not improved over what the cutting plane method alone would produce. In this case, solving the cutting subproblem to obtain additional inequalities is unlikely to yield further improvement.

As with price and cut, there are again many variants of the algorithm shown in Figure 17, depending on the choice of subproblem to execute at each step. One such variant is to alternate between each of the subproblems, first solving one and then the other [45]. In this case, the Lagrangian dual is not solved to optimality before solving the cutting subproblem. Alternatively, another approach is to solve the Lagrangian dual all the way to optimality before generating valid inequalities. Again, the choice is primarily empirical.

Relax and Cut Method

Input: An instance $ILP(\mathcal{P}, c)$.
Output: A lower bound z_{RC} on the optimal solution value for the instance and a dual solution $\hat{u}_{RC} \in \mathbb{R}^{m^t}$.

1. Let $s^0_{RC} \in \mathcal{E}$ define some initial extreme point of $\mathcal{P}'$ and construct an initial outer approximation
$$\mathcal{P}^0_O = \{x \in \mathbb{R}^n \mid D^0 x \geq d^0\} \supseteq \mathcal{P}, \tag{53}$$
where $D^0 = A''$ and $d^0 = b''$. Let $u^0_{RC} \in \mathbb{R}^{m''}$ be some initial set of dual multipliers associated with the constraints $[D^0, d^0]$. Set $t \leftarrow 0$ and $m^t = m''$.

2. **Master Problem:** Using the solution information gained from solving the pricing subproblem, and the previous dual solution u^t_{RC}, update the dual solution (if the pricing problem was just solved) or initialize the new dual multipliers (if the cutting subproblem was just solved) to obtain $u^{t+1}_{RC} \in \mathbb{R}^{m^t}$.

3. Do either (a) or (b).

   (a) **Pricing Subproblem:** Call the subroutine $OPT(\mathcal{P}', c - (u^t_{RC})^\top D^t, (c - (u^t_{RC})^\top D^t) s^t_{RC})$ to obtain
   $$z^t_{RC} = \min_{s \in \mathcal{F}'} \{(c^\top - (u^t_{RC}) D^t) s + d^t (u^t_{RC})\}. \tag{54}$$
   Let $s^{t+1}_{RC} \in \mathcal{E}$ be the optimal solution to this subproblem. Set $[D^{t+1}, d^{t+1}] \leftarrow [D^t, d^t]$, $\mathcal{P}^{t+1}_O \leftarrow \mathcal{P}^t_O$, $m^{t+1} \leftarrow m^t$, $t \leftarrow t+1$, and go to Step 2.

   (b) **Cutting Subproblem**: Call the subroutine $SEP(\mathcal{P}, s^t_{RC})$ to generate a set of *improving* valid inequalities $[\tilde{D}, \tilde{d}] \in \mathbb{R}^{\tilde{m} \times n+1}$ for $\mathcal{P}$, violated by s^t_{RC}. If violated inequalities were found, set $[D^{t+1}, d^{t+1}] \leftarrow \begin{bmatrix} D^t & d^t \\ \tilde{D} & \tilde{d} \end{bmatrix}$ to form a new outer approximation $\mathcal{P}^{t+1}_O$. Set $m^{t+1} \leftarrow m^t + \tilde{m}$, $s^{t+1}_{RC} \leftarrow s^t_{RC}$, $t \leftarrow t+1$, and go to Step 2.

4. If a prespecified stopping criterion is met, then output $z_{RC} = z^t_{RC}$ and $\hat{u}_{RC} = u^t_{RC}$.

5. Otherwise, go to Step 2.

Figure 17: Outline of the relax and cut method

5 Solving the Cutting Subproblem

In this section, we formalize some notions that have been introduced in our examples and provide more details regarding how the cutting subproblem is solved in practice in the context of the various methods we have outlined. We review the well-known *template paradigm* for separation and introduce a new concept called *structured separation*. Finally, we describe a separation algorithm called *decompose and cut* that is closely related to the integrated decomposition methods we have already described and utilizes several of the concepts introduced earlier.

5.1 The Template Paradigm

The ability to generate valid inequalities for $\mathcal{P}$ violated by a given real vector is a crucial step in many of the methods discussed in this paper. Ideally, we would be able to solve the general facet identification problem for $\mathcal{P}$, allowing us to generate a violated valid inequality whenever one exists. This is clearly not practical in most cases, since the complexity of this problem is the same as that of solving the original ILP. In practice, the subproblem $SEP(\mathcal{P}, x^t_{CP})$ in Step 3 of the cutting plane method pictured in Figure 2 is usually solved by dividing the valid inequalities for $\mathcal{P}$ into *template classes* with known structure. Procedures are then designed and executed for identifying violated members of each class individually.

A template class (or simply *class*) of valid inequalities for $\mathcal{P}$ is a set of related valid inequalities that describes a polyhedron containing $\mathcal{P}$, so we can identify each class with its associated polyhedron. In Example 2, we described two well-known classes of valid inequalities for the TSP, the *subtour elimination constraints* and the *comb inequalities*. Both classes have an identifiable coefficient structure and describe polyhedra containing $\mathcal{P}$. Consider a polyhedron $\mathcal{C}$ described by a class of valid inequalities for $\mathcal{P}$. The separation problem for the class $\mathcal{C}$ of valid inequalities for $\mathcal{P}$ is defined to be the facet identification problem over the polyhedron $\mathcal{C}$. In other words, the separation problem for a class of valid inequalities depends on the form of the inequality and is independent of the polyhedron $\mathcal{P}$. It follows that the worst case running time for solving the separation problem is also independent of $\mathcal{P}$. In particular, the separation problem for a particular class of inequalities may be much easier to solve than the general facet identification problem for $\mathcal{P}$. Therefore, in practice, the separation problem is usually attempted over "easy" classes first, and more difficult classes are only attempted when needed. In the case of the TSP, the separation problem for the SECs is solvable in polynomial time, whereas there is no known efficient algorithm for solving the separation problem for comb inequalities. In general, the intersection of the polyhedra associated with the classes of inequalities for which the separation problem can be reasonably solved is not equal to $\mathcal{P}$.

5.2 Separating Solutions with Known Structure

In many cases, the complexity of the separation problem is also affected by the structure of the real vector being separated. In Section 4, we informally introduced the notion that a solution vector with known structure may be easier to separate from a given polyhedron than an arbitrary one and illustrated this phenomenon in Examples 2 and 3. This is a concept called *structured separation* that arises quite frequently in the solution of combinatorial optimization problems where the original formulation is of exponential size. When using the cutting plane method to solve the LP relaxation of the TSP, for example, as described in Example 2, we must generate the SECs dynamically. It is thus possible that the intermediate solutions are integer-valued, but nonetheless not feasible because they violate some SEC that is not present in the current approximation. When the current solution is optimal, however, it is easy to determine whether it violates a SEC by simply examining the connected components of the underlying support graph, as described earlier. This process can be done in $O(|V| + |E|)$ time. For an arbitrary real vector, the separation problem for SECs is more difficult, taking $O(|V|^4)$ time.

It is also frequently the case that when applying a sequence of separation routines for progressively more difficult classes of inequalities, routines for the more difficult classes assume implicitly that the solution to be separated satisfies all inequalities of the the easier classes. In the case of the TSP, for instance, any solution passed to the subroutine for separating the comb inequalities is generally assumed to satisfy the degree and subtour elimination constraints. This assumption can allow the separation algorithms for subsequent classes to be implemented more efficiently.

For the purposes of the present work, our main concern is with separating solutions that are known to be integral, in particular, members of $\mathcal{F}'$. In our framework, the concept of structured separation is combined with the template paradigm in specifying template classes of inequalities for which separation of integral solutions is much easier, in a complexity sense, than separation of arbitrary real vectors over that same class. A number of examples of problems and classes of valid inequalities for which this situation occurs are examined in Section 8. We now examine a separation paradigm called *decompose and cut* that can take advantage of our ability to easily separate solutions with structure.

5.3 Decompose and Cut

The use of a decomposition to aid in separation, as is described in the procedure of Figure 14, is easy to extend to a traditional branch and cut framework using a technique we call *decompose and cut*, originally proposed in [56] and further developed in [39] and [57]. Suppose now that we are given an optimal fractional solution x^t_{CP} obtained during iteration t of the cutting plane method and suppose that for a given $s \in \mathcal{F}'$, we can determine effectively whether $s \in \mathcal{F}$ and if not, generate a valid inequality (a, β) violated by s. By first *decomposing* x^t_{CP} (i.e., expressing x^t_{CP} as a convex combination of members of $\mathcal{E} \subseteq \mathcal{F}'$) and then separating each member of this decomposition from $\mathcal{P}$ in the fashion described in Figure 14, we may be able to find valid inequalities for $\mathcal{P}$ that are violated by x^t_{CP}.

The difficult step is finding the decomposition of x^t_{CP}. This can be accomplished by solving a linear program whose columns are the members of $\mathcal{E}$, as described in Figure 18. This linear program is reminiscent of (33) and in fact can be solved using an analogous column-generation scheme, as described in Figure 19. This scheme can be seen as the "inverse" of the method described in Section 4.1, since it begins with the fractional solution x^t_{CP} and tries to compute a decomposition, instead of the other way around. By the equivalence of optimization and facet identification, we can conclude that the problem of finding a decomposition of x^t_{CP} is polynomially equivalent to that of optimizing over $\mathcal{P}'$.

Separation in Decompose and Cut

Input: $\hat{x} \in \mathbb{R}^n$
Output: 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

$$\max_{\lambda \in \mathbb{R}_+^{\mathcal{E}}} \{\mathbf{0}^\top \lambda \mid \sum_{s \in \mathcal{E}} s\lambda_s = \hat{x}, \sum_{s \in \mathcal{E}} \lambda_s = 1\}, \tag{55}$$

   as in Figure 19.

3. The result of Step 2 is either (1) a subset $\mathcal{D}$ of members of $\mathcal{E}$ participating in a convex combination of $\hat{x}$, or (2) a valid inequality (a, β) 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 18: 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 $\mathcal{E}$ and a vector $\hat{\lambda} \in \mathbb{R}_+^{\mathcal{E}}$ such that $\sum_{s \in \mathcal{D}} \lambda_s s = \hat{x}$ and $\sum_{s \in \mathcal{D}} \lambda_s = 1$.

2.0 Generate an initial subset $\mathcal{E}^0$ of $\mathcal{E}$ and set $t \leftarrow 0$.

2.1 Solve (55), replacing $\mathcal{E}$ by $\mathcal{E}^t$. If this linear program is feasible, then the elements of $\mathcal{E}^t$ corresponding to the nonzero components of $\hat{\lambda}$, the current solution, comprise the set $\mathcal{D}$, so STOP.

2.2 Otherwise, let (a, β) be a valid inequality for $\text{conv}(\mathcal{E}^t)$ violated by $\hat{x}$ (i.e., the proof of infeasibility). Solve $OPT(\mathcal{P}', a, \beta)$ and let $\tilde{\mathcal{E}}$ be the resulting set of solutions. If $\tilde{\mathcal{E}} \neq \emptyset$, then set $\mathcal{E}^{t+1} \leftarrow \mathcal{E}^t \cup \tilde{\mathcal{E}}$, $t \rightarrow t+1$, and go to 2.1. Otherwise, (a, β) is an inequality valid for $\mathcal{P}' \supseteq \mathcal{P}$ and violated by $\hat{x}$, so STOP.

Figure 19: Column generation for the decompose and cut method

Once the decomposition is found, it can be used as before to locate a violated valid inequality. In contrast to price and cut, however, it is possible that $x^t_{CP} \notin \mathcal{P}'$. 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 of Figure 18. The proof of infeasibility for the linear program (55), however, provides an inequality separating x^t_{CP} 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 x^t_{CP}. This idea was originally suggested in [56] and was further developed in [39]. A similar concept was also discovered and developed independently by Applegate, et al. [3].

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 advantage in terms of computational efficiency. In other respects, the computations performed in each method are similar.

6 Solving the Master Problem

The choice of a proper algorithm for solving the master problem is important for these methods, both because a significant portion of the computational effort is spent solving the master problem and because the solver must be capable of returning the solution information required by the method. In this section, we would like to briefly give the reader a taste for the issues involved and summarize the existing methodology. The master problems we have discussed are linear programs, or can be reformulated as linear programs. Hence, one option for solving them is to use either simplex or interior point methods. In the case of solving a Lagrangian dual, subgradient methods may also be employed.

Simplex methods have the advantage of providing accurate primal solution information. They are therefore well-suited for algorithms that utilize primal solution information, such as price and cut. The drawback of these methods is that updates to the dual solution at each iteration are relatively expensive. In their most straightforward implementations, they also tend to converge slowly when used with column generation. This is primarily due to the fact that they produce basic (extremal) dual solutions, which tend to change substantially from one iteration to the next, causing wide oscillations in the input to the column-generation subproblem. This problem can be addressed by implementing one of a number of stabilization methods that prevent the dual solution from changing "too much" from one iteration to the next (for a survey, see [42]).

Subgradient methods, on the other hand, do not produce primal solution information in their most straightforward form, so they are generally most appropriate for Lagrangian methods such as relax and cut. It is possible, however, to obtain approximate primal solutions from variants of subgradient such as the volume algorithm [9]. Subgradient methods also have convergence issues without some form of stabilization. A recent class of algorithms that has proven effective in this regard is bundle methods [18].

Interior point methods may provide a middle ground by providing accurate primal solution information and more stable dual solutions [58, 28]. In addition, hybrid methods that alternate between simplex and subgradient methods for updating the dual solution have also shown promise [10, 36].

7 Software

The theoretical and algorithmic framework proposed in Sections 3–5 lends itself nicely to a wide-ranging and flexible generic software framework. All of the techniques discussed can be implemented by combining a set of basic algorithmic building blocks. DECOMP is a C++ framework designed with the goal of providing a user with the ability to easily utilize various traditional and integrated decomposition methods while requiring only the provision of minimal problem-specific algorithmic components [25]. With DECOMP, the majority of the algorithmic structure is provided as part of the framework, making it easy to compare various algorithms directly and determine which option is the best for a given problem setting. In addition, DECOMP is extensible—each algorithmic component *can* be overridden by the user, if they so wish, in order to develop sophisticated variants of the aforementioned methods.

The framework is divided into two separate user interfaces, an applications interface `DecompApp`, in which the user must provide implementations of problem-specific methods (e.g., solvers for the subproblems), and an algorithms interface `DecompAlgo`, in which the user can modify DECOMP's internal algorithms, if desired. A `DecompAlgo` object provides implementations of all of the methods described in Sections 3 and 4, as well as options for solving the master problem, as discussed in Section 6. One important feature of DECOMP is that the problem is always represented in the original space, rather than in the space of a particular reformulation. The user has only to provide subroutines for separation and column generation in the original space without considering the underlying method. The framework performs all of the necessary bookkeeping tasks, including including automatic reformulation in the Dantzig-Wolfe master, constraint dualization for relax and cut, cut and variable pool management, as well as, row and column expansion.

In order to develop an application, the user must provide implementations of the following two methods.

- `DecompApp::createCore()`. The user must define the initial set of constraints $[A'', b'']$.
- `DecompApp::solveRelaxedProblem()`. The user must provide a solver for the relaxed problem $OPT(\mathcal{P}', c, U)$ that takes a cost vector $c \in \mathbb{R}^n$ as its input and returns a set of solutions as `DecompVar` objects. Alternatively, the user has the option to provide the inequality set $[A', b']$ and solve the relaxed problem using the built-in ILP solver.

If the user wishes to invoke the traditional cutting plane method using problem-specific methods, then the following method must also be implemented.

- `DecompApp::generateCuts(`x`)`. A method for solving the separation problem $SEP(\mathcal{P}, x)$, given an arbitrary real vector $x \in \mathbb{R}^n$, which returns a set of `DecompCut` objects.

Alternatively, various generic separation algorithms are also provided. The user might also wish to implement separation routines specifically for members of $\mathcal{F}'$ that can take advantage of the structure of such solutions, as was described in Section 5.

- `DecompApp::generateCuts(`s`)`. A method for solving the separation problem $SEP(\mathcal{P}, s)$, given $s \in \mathcal{F}'$, which returns a set of `DecompCut` objects.

At a high level, the main loop of the base algorithm provided in `DecompAlgo` follows the paradigm described earlier, alternating between solving a master problem to obtain solution information, followed by a subproblem to generate new polyhedral information. Each of the methods described in this paper have its own separate interface derived from `DecompAlgo`. For example, the base class for the price and cut method is `DecompAlgo::DecompAlgoPC`. In this manner, the user can override a specific subroutine common to all methods (in `DecompAlgo`) or restrict it to a particular method.

8 Applications

In this section, we further illustrate the concepts presented with three more examples. We focus here on the application of integrated methods, a key component of which is the paradigm of structured separation introduced in Section 5. For each example, we discuss three key polyhedra: (1) an original ILP defined by a polyhedron $\mathcal{P}$ and associated feasible set $\mathcal{F} = \mathcal{P} \cap \mathbb{Z}^n$; (2) a relaxation of the original ILP with feasible set $\mathcal{F}' \supseteq \mathcal{F}$ such that effective optimization over the polyhedron $\mathcal{P}_I = \text{conv}(\mathcal{F}')$ is possible; and (3) a polyhedron $\mathcal{P}_O$, such that $\mathcal{F} = \mathcal{P}_I \cap \mathcal{P}_O \cap \mathbb{Z}^n$. In each case, the polyhedron $\mathcal{P}_O$ is comprised of a known class or classes of valid inequalities that could be generated during execution of the cutting subproblem of one of the integrated methods discussed in Section 4. As before, $\mathcal{P}_I$ is a polyhedron with an inner description generated dynamically through the solution of an optimization problem, while $\mathcal{P}_O$ is a polyhedron with an outer description generated dynamically through the solution of a separation problem. We do not discuss standard methods of solving the separation problem for $\mathcal{P}_O$, i.e., unstructured separation, as these are well-covered in the literature. Instead, we focus here on problems and classes of valid inequalities for which structured separation, i.e., separation of a member of $\mathcal{F}'$, is much easier than unstructured separation. A number of ILPs that have appeared in the literature have relaxations and associated classes of valid inequalities that fit into this framework, such as the Generalized Assignment Problem [54], the Edge-Weighted Clique Problem [37], the Knapsack Constrained Circuit Problem [41], the Rectangular Partition Problem [16], the Linear Ordering Problem [15], and the Capacitated Minimum Spanning Tree Problem [24].

8.1 Vehicle Routing Problem

We first consider the *Vehicle Routing Problem* (VRP) introduced by Dantzig and Ramser [20]. 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 $V = \{1, \ldots, |V|\}$ 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 of 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 vertex set $N = V \cup \{0\}$ and edge set E, we can formulate the VRP as an integer program. A *route* is a set of vertices $R = \{i_1, i_2, \ldots, i_m\}$ such that the members of R are distinct. The edge set of R

is $E_R = \{\{i_j, i_{j+1}\} \mid j \in 0, \ldots, m\}$, where $i_0 = i_{m+1} = 0$. 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 of which satisfies 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. To simplify the presentation, let us define some additional notation.

By associating a variable with each edge in the graph, we obtain the following formulation of this ILP [40]:

$$
\begin{aligned}
\min \sum_{e \in E} c_e x_e,& & & \\
x(\delta(\{0\})) &= 2k, & & (56)\\
x(\delta(\{v\})) &= 2 & \forall v \in V, & \quad (57)\\
x(\delta(S)) &\geq 2b(S) & \forall S \subseteq V,\ |S| > 1, & \quad (58)\\
x_e &\in \{0,1\} & \forall e \in E(V), & \quad (59)\\
x_e &\in \{0,1,2\} & \forall e \in \delta(0). & \quad (60)
\end{aligned}
$$

Here, $b(S)$ represents a lower bound on the number of vehicles required to service the set of customers S. Inequalities (56) ensure that there are exactly k vehicles, each departing from and returning to the depot, while inequalities (57) require that each customer must be serviced by exactly one vehicle. Inequalities (58), known as the *generalized subtour elimination constraints* (GSECs) can be viewed as a generalization of the subtour elimination constraints from 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.

The set of feasible solutions to the VRP is

$$
\mathcal{F} = \{x \in \mathbb{R}^E \mid x \text{ satisfies } (56) - (60)\}
$$

and we call $\mathcal{P} = \text{conv}(\mathcal{F})$ the *VRP polyhedron.* Many classes of valid inequalities for the VRP polyhedron have been reported in the literature (see [51] for a survey). Significant effort has been devoted to developing efficient algorithms for separating an arbitrary fractional point using these classes of inequalities (see [46] for recent results).

We concentrate here on the separation of GSECs. The separation problem for GSECs was shown to be $\mathcal{NP}$-complete by Harche and Rinaldi (see [5]), even when $b(S)$ is taken to be $\left\lceil \left(\sum_{i \in S} d_i\right)/C \right\rceil$. In [46], Lysgaard, et al. review heuristic procedures for generating violated GSECs. Although GSECs are part of the formulation presented above, there are exponentially many of them, so we generate them dynamically. We discuss three relaxations of the VRP: the *Multiple Traveling Salesman Problem*, the *Perfect b-Matching Problem*, and the *Minimum Degree-constrained k-Tree Problem.* For each of these alternatives, violation of GSECs by solutions to the relaxation can be easily discerned.

Perfect b-Matching Problem. With respect to the graph G, the *Perfect b-Matching Problem* is to find a minimum weight subgraph of G such that $x(\delta(v)) = b_v \; \forall v \in V$. This problem can be formulated by dropping the GSECs from the VRP formulation, resulting in the feasible set

$$\mathcal{F}' = \{x \in \mathbb{R}^E \mid x \text{ satisfies } (56), (57), (59), (60)\}.$$

In [50], Müller-Hannemann and Schwartz, present several fast polynomial algorithms for solving b-Matching. The polyhedron $\mathcal{P}_O$ consists of the GSECs (58) in this case.

In [49], Miller uses the b-Matching relaxation to solve the VRP by branch, relax, and cut. He suggests generating GSECS violated by b-matchings as follows. Consider a member s of $\mathcal{F}'$ and its support graph G_s (a b-Matching). If G_s is disconnected, then each component immediately induces a violated GSEC. On the other hand, if G_s is connected, we first remove the edges incident to the depot vertex and find the connected components, which comprise the routes described earlier. To identify a violated GSEC, we 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 on any route whose total demand exceeds capacity induces a violated GSEC. This separation routine runs in $O(|V| + |E|)$ time and can be used in any of the integrated decomposition methods previously described. Figure 20(a) shows an example vector that could arise during execution of either price and cut or decompose and cut, along with a decomposition into a convex combination of two b-Matchings, shown in Figures 20(b) and 20(c). In this example, the capacity $C = 35$ and by inspection we find a violated GSEC in the second b-Matching (c) with S equal to the marked component. This inequality is also violated by the optimal fractional solution, since $\hat{x}(\delta(S)) = 3.0 < 4.0 = 2b(S)$.

Minimum Degree-constrained k-Tree Problem. A *k-tree* is defined as a spanning subgraph of G that has $|V|+k$ edges (recall that G has $|V|+1$ vertices). A *degree-constrained k-tree* (k-DCT), as defined by Fisher in [23], is a k-tree with degree $2k$ at vertex 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. Fisher [23] introduced this relaxation as part of a Lagrangian relaxation-based algorithm for solving the VRP.

The k-DCT polyhedron is obtained by first adding the redundant constraint

$$x(E) = |V| + k, \tag{61}$$

then deleting the degree constraints (57), and finally, relaxing the capacity to $C = \sum_{i \in S} d_i$. Relaxing the capacity constraints gives $b(S) = 1$ for all $S \subseteq V$, and replaces the set of constraints (58) with

$$\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subseteq V, \; |S| > 1. \tag{62}$$

The feasible region of the Minimum k-DCT Problem is then

$$\mathcal{F}' = \{x \in \mathbb{R}^E \mid x \text{ satisfies } (56), (58), (59), (61)\}.$$

This time, the polyhedron $\mathcal{P}_O$ is comprised of the constraints (57) and the GSECs (58). Since the constraints (57) can be represented explicitly, we focus again on generation of violated GSECs. In [62], Wei and Yu give a polynomial algorithm for solving the Minimum k-DCT Problem that runs in $O(|V|^2 \log |V|)$ time. In [48], Martinhon et al. study the use of the k-DCT relaxation for the VRP in the context branch, relax, and cut. Again, consider separating a member s of $\mathcal{F}'$ from the polyhedron defined by all GSECS. It is easy to see that for GSECs, an algorithm identical to that described above can be applied. Figure 20(a) also shows a vector that could arise during the execution of either the price and cut or decompose and cut algorithms, along with a decomposition into a convex combination of four k-DCTs, shown in Figures 20(d) through 20(g). Removing the depot edges and checking each component's demand, we easily identify the violated GSEC shown in Figure 20(g).

Multiple Traveling Salesman Problem. The *Multiple Traveling Salesman Problem* (k-TSP) is an uncapacitated version of the VRP obtained by adding the degree constraints to the k-DCT polyhedron. The feasible region of the k-TSP is

$$\mathcal{F}' = \{x \in \mathbb{R}^E \mid x \text{ satisfies } (56), (57), (59), (60), (62)\}.$$

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 vertex 0 and its incident edges. In this case, the polyhedron $\mathcal{P}_O$ is again comprised solely of the GSECs (58). In [57], Ralphs et al. report on an implementation of branch, decompose and cut using the k-TSP as a relaxation.

8.2 Three-Index Assignment Problem

The *Three-Index Assignment Problem* (3AP) 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 $H = I \times J \times K$. 3AP can be formulated as the following binary integer program:

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

$$\sum_{(j,k) \in J \times K} x_{ijk} = 1 \qquad \forall i \in I, \tag{63}$$

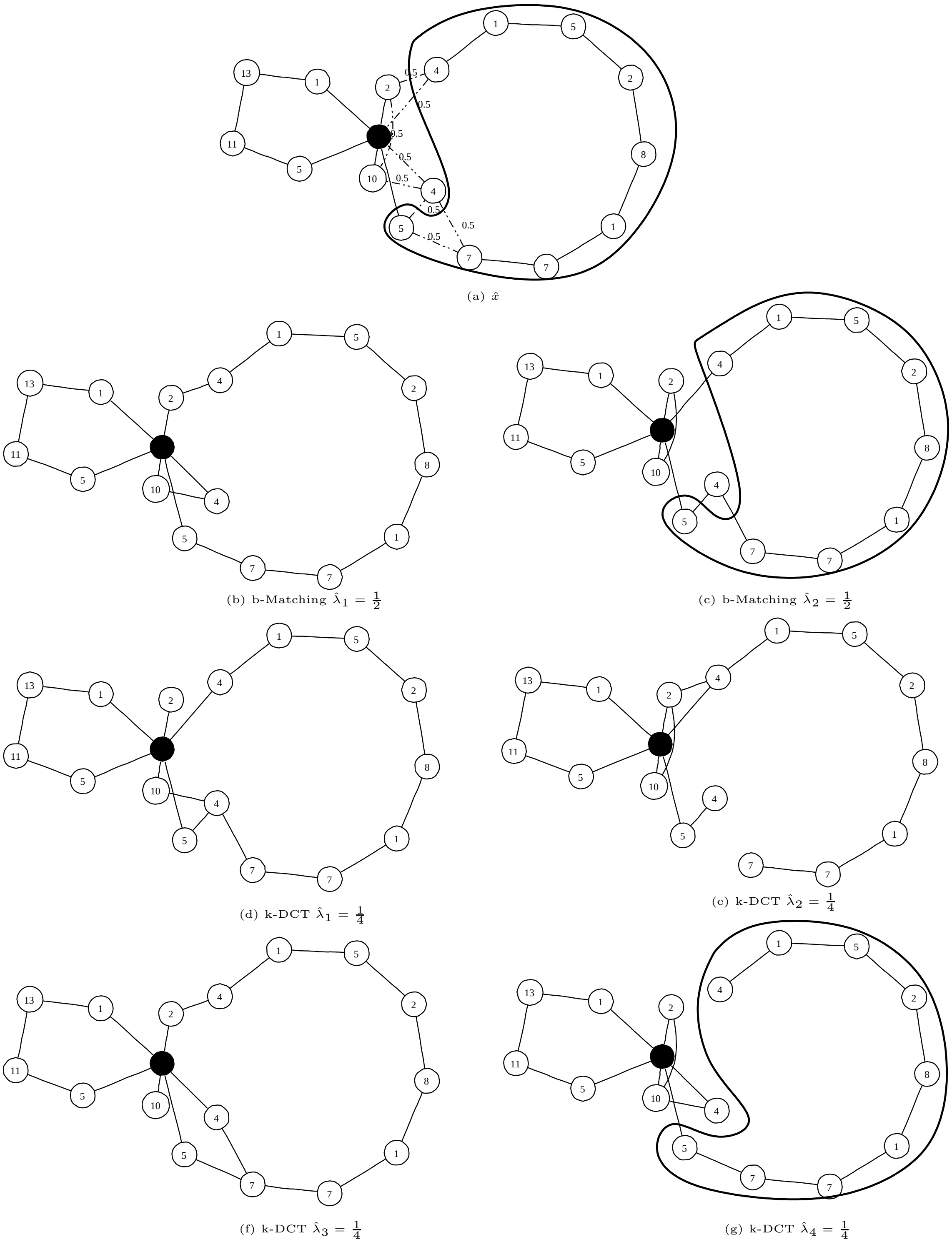
$$\sum_{(i,k) \in I \times K} x_{ijk} = 1 \qquad \forall j \in J, \tag{64}$$

$$\sum_{(i,j) \in I \times J} x_{ijk} = 1 \qquad \forall k \in K, \tag{65}$$

$$x_{ijk} \in \{0,1\} \ \forall (i,j,k) \in H. \tag{66}$$

A number of applications of 3AP can be found in the literature (see Piersjalla [18,19]). 3AP is known to be $\mathcal{NP}$-hard [26]. As before, the set of feasible solutions to 3AP is noted as

$$\mathcal{F} = \{x \in \mathbb{R}^H \mid x \text{ satisfies } (63) - (66)\}$$



(a) $\hat{x}$

(b) b-Matching $\hat{\lambda}_1 = \frac{1}{2}$

(c) b-Matching $\hat{\lambda}_2 = \frac{1}{2}$

(d) k-DCT $\hat{\lambda}_1 = \frac{1}{4}$

(e) k-DCT $\hat{\lambda}_2 = \frac{1}{4}$

(f) k-DCT $\hat{\lambda}_3 = \frac{1}{4}$

(g) k-DCT $\hat{\lambda}_4 = \frac{1}{4}$

Figure 20: Example of a decomposition into b-Matchings and k-DCTs

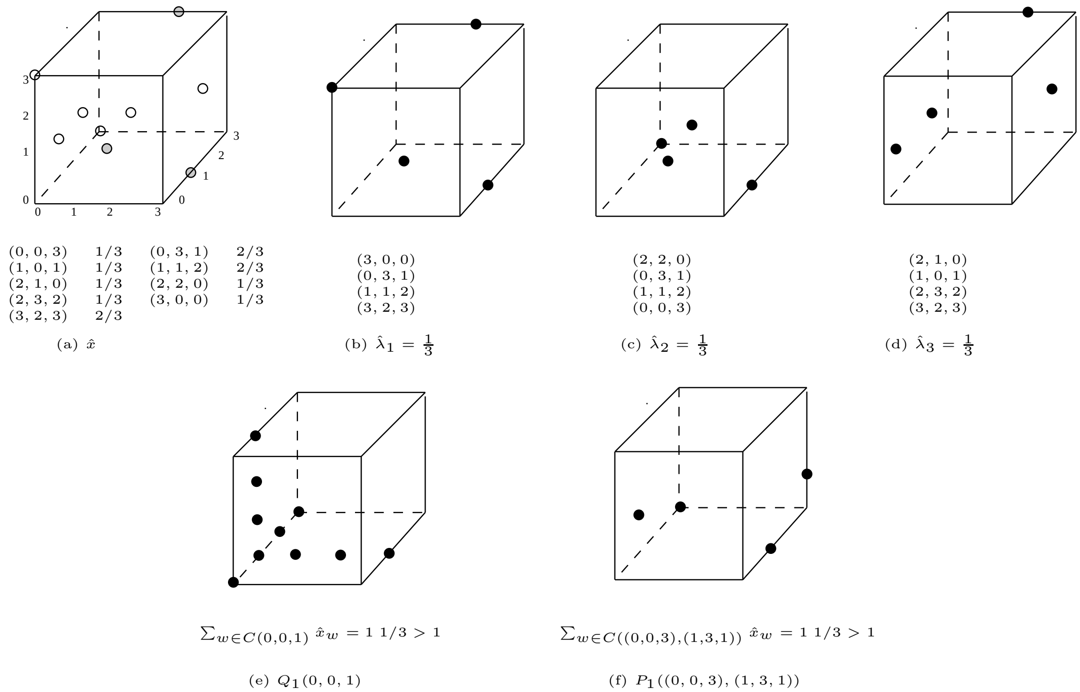


Figure 21: Example of a decomposition into assignments.

and we set $\mathcal{P} = \text{conv}(\mathcal{F})$.

In [7], 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 coordinates for which the vectors u and v have the same value. Let $C(u) = \{w \in H \mid |u \cap w| = 2\}$ and $C(u, v) = \{w \in H \mid |u \cap w| = 1, |w \cap v| = 2\}$. We consider two classes of facet-inducing inequalities $Q_1(u)$ and $P_1(u, v)$ for $\mathcal{P}$,

$$x_u + \sum_{w \in C(u)} x_w \leq 1 \ \forall u \in H, \tag{67}$$

$$x_u + \sum_{w \in C(u,v)} x_w \leq 1 \ \forall u, v \in H, |u \cap v| = 0. \tag{68}$$

Note that these include all the clique facets of the intersection graph of $K_{n,n,n}$ [7]. In [6], Balas and Qi describe algorithms that solve the separation problem for the polyhedra defined by the inequalities $Q_1(u)$ and $P_1(u, v)$ in $O(n^3)$ time.

Balas and Saltzman consider the use of the classical *Assignment Problem* (AP) as a relaxation of 3AP in an early implementation of branch, relax, and cut [8]. The feasible region of the AP is

$$\mathcal{F}' = \{x \in \mathbb{R}^H \mid x \text{ satisfies } (64) - (66)\}.$$

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 [2]. The polyhedron $\mathcal{P}_O$ is here described by constraints (63), the constraints

$Q_1(u)$ for all $u \in H$, and the constrains $P_1(u,v)$ for all $u, v \in H$. Consider generation of a constraint of the form $Q_1(u)$ for some $u \in H$ violated by a given $s \in \mathcal{F}'$. 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 21 shows the decomposition of a vector $\hat{x}$ (a) the could arise during the execution of either the price and cut or decompose and algorithms, along with a decomposition of $\hat{x}$ into a convex combination of assignments (b-d). The pair of triplets $(0, 3, 1)$ and $(0, 0, 3)$ satisfies the condition just discussed and identifies two valid inequalities, $Q_1(0, 3, 3)$ and $Q_1(0, 0, 1)$, that are violated by the second assignment, shown in (c). The latter also violates $\hat{x}$ and is illustrated in (e). This separation routine runs in $O(n)$ time.

Now consider generation of a constraint of the form $P_1(u, v)$ for some $u, v \in H$ violated by $s \in \mathcal{F}'$. 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 $P_1((i_0, j_1, k_1), (i, j_0, k_0)), \forall i \neq i_0$. The inequality $P_1((0, 0, 3), (1, 3, 1))$ is violated by the second assignment, shown in Figure 21(c). This inequality is also violated by $\hat{x}$ and is illustrated in (f). Once again, this separation routine runs in $O(n)$ time.

8.3 Steiner Tree Problem

Let $G = (V, E)$ be a complete undirected graph with vertex set $V = \{1, ..., |V|\}$, edge set E and a positive weight c_e associated with each edge $e \in E$. Let $T \subset V$ define the set of *terminals*. The *Steiner Tree Problem* (STP), which is $\mathcal{NP}$-hard, is that of finding a subgraph that spans T (called a *Steiner tree*) and has minimum edge cost. In [13], Beasley formulated the STP as a side constrained *Minimum Spanning Tree Problem* (MSTP) as follows. Let $r \in T$ be some terminal and define an artificial vertex 0. Now, construct the augmented graph $\bar{G} = (\bar{V}, \bar{E})$ where $\bar{V} = V \cup \{0\}$ and $\bar{E} = E \cup \{\{i, 0\} \mid i \in (V \setminus T) \cup \{r\}\}$. Let $c_{i0} = 0$ for all $i \in (V \setminus T) \cup \{r\}$. Then, the STP is equivalent to finding a minimum spanning tree (MST) in $\bar{G}$ subject to the additional restriction that any vertex $i \in (V \setminus T)$ connected by edge $\{i, 0\} \in \bar{E}$ must have degree one.

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

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

$$x(\bar{E}) = |\bar{V}| - 1, \tag{69}$$

$$x(E(S)) \leq |S| - 1 \quad \forall S \subseteq \bar{V}, \tag{70}$$

$$x_{i0} + x_e \leq 1 \quad \forall e \in \delta(i), i \in (V \setminus T), \tag{71}$$

$$x_e \in \{0, 1\} \quad \forall e \in \bar{E}. \tag{72}$$

Inequalities (69) and (70) ensure that the solution forms a spanning tree on $\bar{G}$. Inequalities (70) are subtour elimination constraints (similar to those used in the TSP). Inequalities

(71) are the side constraints that ensure the solution can be converted to a Steiner tree by dropping the edges in $\bar{E} \setminus E$.

The set of feasible solutions to the STP is

$$\mathcal{F} = \{x \in \mathbb{R}^{\bar{E}} \mid x \text{ satisfies } (69) - (72)\}.$$

We set $\mathcal{P} = \text{conv}(\mathcal{F})$ as before. We consider two classes of valid inequalities for $\mathcal{P}$ that are lifted versions of the subtour elimination constraints (SEC).

$$x(E(S)) + x(E(S \setminus T \mid \{0\})) \leq |S| - 1 \;\; \forall S \subseteq V, S \cap T \neq \emptyset, \tag{73}$$

$$x(E(S)) + x(E(S \setminus \{v\} \mid \{0\})) \leq |S| - 1 \;\; \forall S \subseteq V, S \cap T = \emptyset, v \in S. \tag{74}$$

The class of valid inequalities (73) were independently introduced by Goemans [27], Lucena [43] and Margot, Prodon, and Liebling [47], for another extended formulation of STP. The inequalities (74) were introduced in [27, 47]. The separation problem for inequalities (73) and (74) can be solved in $O(|V|^4)$ time through a series of max-flow computations.

In [44], Lucena considers the use of MSTP as a relaxation of STP in the context of a branch, relax, and cut algorithm. The feasible region of the MSTP is

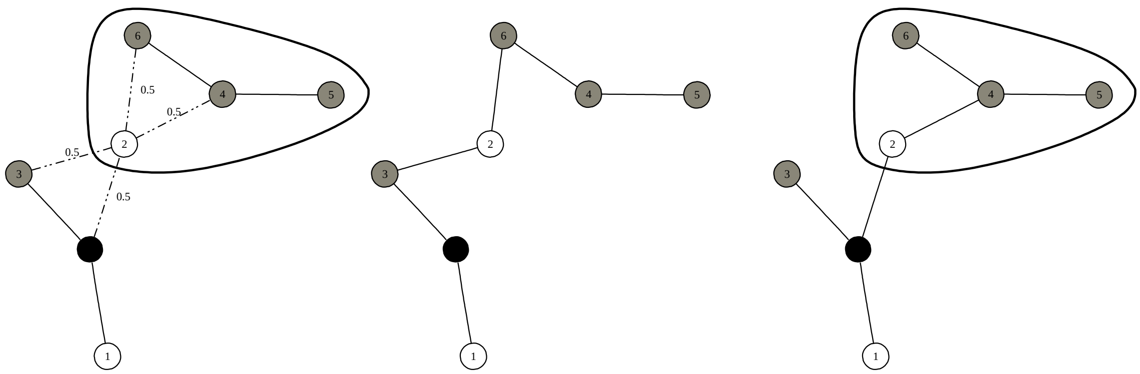
$$\mathcal{F}' = \{x \in \mathbb{R}^{\bar{E}} \mid x \text{ satisfies } (69), (70), (72)\}.$$

The MSTP can be solved in $O(|E| \log |V|)$ time using Prim's algorithm [55]. The polyhedron $\mathcal{P}_O$ is described by the constraints (71), (73), and (74). Constraints (71) can be represented explicitly, but we must dynamically generate constraints (73) and (74). In order to identify an inequality of the form (73) or (74) violated by a given $s \in \mathcal{F}'$, we remove the artificial vertex 0 and find the connected components on the resulting subgraph. Any component of size greater than 1 that does not contain r and does contain a terminal, defines a violated SEC (73). In addition, if the component does not contain any terminals, then each vertex in the component that was not connected to the artificial vertex defines a violated SEC (74).

Figure 22 gives an example of a vector (a) that might have arisen during execution of either the price and cut or decompose and cut algorithms, along with a decomposition into a convex combination of two MSTs (b,c). In this figure, the artificial vertex is black, the terminals are gray and $r = 3$. By removing the artificial vertex, we easily find a violated SEC in the second spanning tree (c) with S equal to the marked component. This inequality is also violated by the optimal fractional solution, since $\hat{x}(E(S)) + \hat{x}(E(S \setminus T \mid \{0\})) = 3.5 > 3 = |S| - 1$. It should also be noted that the first spanning tree (b), in this case, is in fact feasible for the original problem.

9 Conclusions and Future Work

In this paper, we presented a framework for integrating dynamic cut generation (outer methods) and traditional decomposition methods (inner methods) to yield new integrated methods that may produce bounds that are improved over those yielded by either technique alone. We showed the relationships between the various methods and how they can



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

Figure 22: Example of a decomposition into minimum spanning trees.

be viewed in terms of polyhedral intersection. We have also introduced the concept of structured separation and a related paradigm for the generation of improving inequalities based on decomposition and the separation of solutions to a relaxation. The next step in this research is to complete a computational study using the software framework introduced in Section 7 that will allow practitioners to make intelligent choices between the many possible variants we have discussed.

References

[1] K. Aardal and S. van Hoesel. Polyhedral techniques in combinatorial optimization. *Statistica Neerlandica*, 50:3–26, 1996.

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

[3] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. TSP cuts which do not conform to the template paradigm. In *Computational Combinatorial Optimization*, pages 261–303. Springer, 2001.

[4] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems. *Mathematical Programming*, 97:91–153, 2003.

[5] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Technical Report 949-M, Université Joseph Fourier, Grenoble, France, 1995.

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

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

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

[9] F. Barahona and R. Anbil. The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming*, 87:385–399, 2000.

[10] F. Barahona and D. Jensen. Plant location with minimum inventory. *Mathematical Programming*, 83:101–111, 1998.

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

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

[13] J.E. Beasley. A SST-based algorithm for the steiner problem in graphs. *Networks*, 19:1–16, 1989.

[14] J.E. Beasley. Lagrangean relaxation. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Optimization*. Wiley, 1993.

[15] A. Belloni and A. Lucena. A Lagrangian heuristic for the linear ordering problem. In M.G.C. Resende and J. Pinho de Sousa, editors, *Metaheuristics: Computer Decision-Making*, pages 123–151. Kluwer Academic, 2003.

[16] F. C. Calheiros, A. Lucena, and C. de Souza. Optimal rectangular partitions. *Networks*, 41:51–67, 2003.

[17] A. Caprara, M. Fischetti, and P. Toth. Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371, 2000.

[18] P. Carraresi, A. Frangioni, and M. Nonato. Applying bundle methods to optimization of polyhedral functions: An applications-oriented development. Technical Report TR-96-17, Universitá di Pisa, 1996.

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

[20] G.B. Danzig and R.H. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

[21] M. Poggi de Aragão and E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price. Working paper, Pontifíca Universidade Católica do Rio de Janeiro, 2004. Available from http://www.inf.puc-rio.br/˜uchoa/doc/rbcp-a.pdf.

[22] M.L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27:1–18, 1981.

[23] M.L. Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42:626–642, 1994.

[24] R. Fukasawa, M. Poggi de Aragão, M. Reis, and E. Uchoa. Robust branch-and-cut-and-price for the capacitated minimum spanning tree problem. In *Proceedings of the International Network Optimization Conference*, pages 231–236, Evry, France, 2003.

[25] M. Galati. DECOMP user's manual. Technical report, Lehigh University Industrial and Systems Engineering, 2005.

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

[27] M.X. Goemans. The steiner tree polytope and related polyhedra. *Mathematical Programming*, 63:157–182, 1994.

[28] J. L. Goffin and J. P. Vial. Convex nondifferentiable optimization: A survey focused on the analytic center cutting plane method. Technical Report 99.02, Logilab, Geneva, Switzerland, 1999.

[29] J. Gondzio and R. Sarkissian. Column generation with a primal-dual method. Technical report, University of Geneva, Logilab, HEC Geneva, 1996.

[30] M. Grötschel and M. Padberg. On the symmetric travelling salesman problem I: Inequalities. *Mathematical Programming*, 16:265–280, 1979.

[31] M. Grötschel and M. Padberg. On the symmetric travelling salesman problem II: Lifting theorems and facets. *Mathematical Programming*, 16:281–302, 1979.

[32] Grötschel, M. and Lovász, L. and Schrijver, A. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[33] M. Guignard. Efficient cuts in Lagrangean "relax-and-cut" schemes. *European Journal of Operational Research*, 105:216–223, 1998.

[34] M. Guignard. Lagrangean relaxation. *Top*, 11:151–228, 2003.

[35] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[36] D. Huisman, R. Jans, M. Peeters, and A. Wagelmans. Combining column generation and Lagrangian relaxation. Technical report, Erasmus Research Institute of Management, Rotterdamn, The Netherlands, 2003.

[37] M. Hunting, U. Faigle, and W. Kern. A Lagrangian relaxation approach to the edge-weighted clique problem. *European Journal of Operational Research*, 131:119–131, 2001.

[38] N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, and F. Soumis. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33:101–116, 1999.

[39] L. Kopman. *A New Generic Separation Routine and Its Application In a Branch and Cut Algorithm for the Capacitated Vehicle Routing Problem*. PhD thesis, Cornell University, May 1999.

[40] G. Laporte, Y. Nobert, and M. Desrouchers. Optimal routing with capacity and distance restrictions. *Operations Research*, 33:1050–1073, 1985.

[41] J. Linderoth and M. Galati. Knapsack constrained circuit problem. Unpublished working paper, 2004.

[42] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. Technical Report 008-2004, Technische Universität Berlin, 2004.

[43] A. Lucena. Tight bounds for the steiner problem in graphs. Talk given at TIMS-XXX-SOBRAPO XXIII Joint International Meeting, Rio de Janeiro, July 1991.

[44] A. Lucena. Steiner problem in graphs: Lagrangean relaxation and cutting planes. *COAL Bulletin*, 28:2–8, 1992.

[45] A. Lucena. Non-delayed relax-and-cut algorithms. Working paper, Departmaneto de Administração, Universidade Federal do Rio de Janeiro, 2004.

[46] J. Lysgaard, A.N. Letchford, and R.W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100:423–445, 2004.

[47] F. Margot, A. Prodon, and T.M. Liebling. Tree polyhedron on 2-tree. *Mathematical Programming*, 63:183–192, 1994.

[48] C. Martinhon, A. Lucena, and N. Maculan. Stronger k-tree relaxations for the vehicle routing problem. *European Journal of Operational Research*, 158:56–71, 2004.

[49] D. Miller. A matching based exact algorithm for capacitated vehicle routing problems. *ORSA Journal on Computing*, 7:1–9, 1995.

[50] M. Müller-Hannemann and A. Schwartz. Implementing weighted b-matching algorithms: Towards a flexible software design. In *Proceedings of the Workshop on Algorithm Engineering and Experimentation (ALENEX99)*, volume 1619 of *Lecture notes in computer science*, pages 18–36, Baltimore, MD, 1999. Springer-Verlag.

[51] D. Naddef and G. Rinaldi. Branch-and-cut algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 53–84. SIAM, 2002.

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

[53] M. Padberg and G. Rinaldi. An efficient algorithm for the minimum capacity cut problem. *Mathematical Programming*, 47:19–36, 1990.

[54] A. Pigatti. *Modelos e algoritmos para o problema de aloção generalizada e aplicações.* PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2003.

[55] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[56] T.K. Ralphs. *Parallel Branch and Cut for Vehicle Routing.* PhD thesis, Cornell University, May 1995.

[57] T.K. Ralphs, L. Kopman, W.R. Pulleyblank, and L.E. Trotter Jr. On the capacitated vehicle routing problem. *Mathematical Programming*, 94:343–359, 2003.

[58] L. M. Rousseau, M. Gendreau, and D. Feillet. Interior point stabilization for column generation. Working paper, Cahier du Gerad, 2003. Available from `http://www.lia.univ-avignon.fr/fich_art/380-IPS.pdf`.

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

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

[61] F. Vanderbeck. Lot-sizing with start-up times. *Management Science*, 44:1409–1425, 1998.

[62] G. Wei and G. Yu. An improved $O(n^2 \log n)$ algorithm for the degree-constrained minimum k-tree problem. Technical report, The University of Texas at Austin, Center for Management of Operations and Logistics, 1995.