Fractional Decomposition Trees: A tool for studying mixed-integer-program integrality gaps

Robert D. Carr[1], Arash Haddadan[2], and Cynthia A. Phillips[3]

[1] University of New Mexico, bobcarr@unm.edu
[2] Carnegie Mellon University, ahaddada@cmu.edu
[3] Sandia National Laboratories, caphill@sandia.gov

Abstract. We present a new algorithm/tool for studying integrality gaps of mixed-integer linear programming formulations. The algorithm is based on convex decomposition of scaled linear-programming relaxations. The relationship between convex decomposition and integrality gaps provides both integrality-gap information and approximate solutions. Our algorithm runs in polynomial time and is guaranteed to find a feasible integer solution provided the integrality gap is bounded. Thus when the algorithm fails, it proves an unbounded integrality gap. The algorithm also provides a lower bound on the instance integrality gap at each step. We apply our algorithm to a class of fractional extreme points for two traveling-salesman-like problems: 2-edge-connected spanning subgraph (2EC) and tree augmentation. These experiments provide insight into the current gap bounds. Furthermore, for 2EC, the approximate solutions are consistently better than the best previous approximation algorithm due to Christofides.

Keywords: Mixed-integer linear programming · Integrality gap · convex combinations.

1 Introduction

Mixed-integer linear programming (MILP), the optimization of a linear objective function subject to linear and integrality constraints, models many practical optimization problems including scheduling, logistics and resource allocation. The set of feasible points for a MILP is the set

$$S(A,G,b) = \{(x,y) \in \mathbb{Z}^n \times \mathbb{R}^p \ : \ Ax + Gy \geq b\}. \tag{1}$$

If we drop the integrality constraints, we have the linear relaxation of set $S(A,G,b)$,

$$P(A,G,b) = \{(x,y) \in \mathbb{R}^{n+p} \ : \ Ax + Gy \geq b\}. \tag{2}$$

Let $I = (A,G,b)$ be the feasible set of a specific instance. Then $S(I)$ and $P(I)$ denote $S(A,G,b)$ and $P(A,G,b)$, respectively. Given a linear objective function c, a MILP problem is min $\{cx : \ (x,y) \in S(I)\}$. It is NP-hard even to determine if a MILP instance has a feasible solution [11]. However, intelligent branch-and-bound strategies allow commercial and open-source MILP solvers to give exact solutions (or near-optimal with provable bound) to many specific instances of NP-hard combinatorial optimization problems.

Relaxing the integrality constraints gives the polynomial-time-solvable linear-programming relaxation: min $\{cx : (x,y) \in P(I)\}$. The optimal value of this linear program (LP), denoted $z_{LP}(I,c)$, is a lower bound on the optimal value for the MILP, denoted $z_{IP}(I,c)$. The solution can also provide some useful global structure, even though the fractional values are not directly meaningful. *LP-based approximation algorithms* for combinatorial problems involve modeling the problem as a MILP, solving the LP relaxation, finding a (problem-specific) integer-feasible solution from the LP solution, and proving an approximation bound by comparing the solution value to the LP lower bound.

Many researchers (see [17,18]) have developed polynomial-time LP-based algorithms that find solutions for special classes of MILPs whose cost are provably smaller than $C \cdot z_{LP}(I,c)$. The approximation factor C can be a constant or depend a MILP parameter, e.g. $O(\log(n))$. However, for many combinatorial optimization problems there is a limit to such techniques. Define the *integrality gap* of the MILP formulation for instance I to be $g_I = \max_{c\geq 0} \frac{z_{IP}(I,c)}{z_{LP}(I,c)}$. This value depends on the constraints in (1). We cannot hope to find solutions for the MILP with objective values better than $g_I \cdot z_{LP}(I,c)$.

More generally we can define the integrality gap for a class of instances $\mathcal{I}$:

$$g_{\mathcal{I}} = \max_{c\geq 0, I\in\mathcal{I}} \frac{z_{IP}(I,c)}{z_{LP}(I,c)} \tag{3}$$

For example, finding a minimum-weight 2-edge-connected multigraph has a natural formulation: every cut is crossed at least twice. The gap for this formulation is at most $\frac{3}{2}$ [19] and at least $\frac{6}{5}$ [2]. Therefore, we cannot hope to obtain an LP-based $(\frac{6}{5}-\epsilon)$-approximation algorithm for this problem using this LP relaxation.

The value of good MILP formulations: There can be multiple correct MILP formulations for a problem with different integrality gaps. Finding MILP formulations with small integrality gap, e.g. by adding extra contraints, enables better provable approximation algorithms. Such formulations are also likely to work better in practice when using exact solvers because branch-and-bound algorithms for MILP use LP bounds to prove whole regions of the search space can be pruned. In this paper, we provide tools to help modelers develop MILP formulations with integrality gaps closer to the optimal.

Decomposition Our methods apply theory connecting integrality gaps to sets of feasible solutions. Instances I with $g_I = 1$ has $P(I) = \mathrm{conv}(S(I))$, the convex hull of the lattice of feasible points. In this case, $P(I)$ is an *integral* polyhedron. The spanning-tree polytope and the perfect-matching polytope [16] have this property. For such problems there is an algorithm to express vector $x \in P(I)$ as a convex combination of points in $S(I)$ in polynomial time [13].

Proposition 1. *If $g_I = 1$, then for $(x,y) \in P(I)$, there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $(\tilde{x}^i, \tilde{y}^i) \in S(I)$ for $i = 1,\ldots,k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq x$. Moreover, we can find such a convex combination in polynomial time.*

Carr and Vempala [3] gave a decomposition result for integrality gap $1 < g(I) < \infty$. This is a generalization of Goemans' proof for blocking polyhedra [12].

Theorem 1 (Carr, Vempala [3]). *Let $(x, y) \in P(I)$, there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $(\tilde{x}^i, \tilde{y}^i) \in \mathcal{D}(S(I))$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq Cx$ if and only if $g_I \leq C$.*

Here $\mathcal{D}(P(I))$ is the set of points (x', y') such that there exists a point $(x, y) \in P$ with $x' \geq x$, also known as the dominant of $P(I)$. For covering problems the polyhedron is essentially the same as its dominant, but this is not true in general. While there is an exact algorithm for problems with gap 1, Theorem 1 is existenial, with no construction. To study integrality gaps, we wish to find such a solution contructively:

Question 1. Assume reasonable complexity assumptions, a specific problem $\mathcal{I}$ with $1 < g_{\mathcal{I}} < \infty$, and $(x, y) \in P(I)$ for some $I \in \mathcal{I}$, can we find $\theta \in [0,1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $(\tilde{x}^i, \tilde{y}^i) \in S(I)$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq$C$g_{\mathcal{I}} x$ in polynomial time? We wish to find the smallest slack factor C as possible.

We give a general approximation framework for solving $\{0,1\}$-MILPs. Consider the set of point described by sets $S(I)$ and $P(I)$ as in (1) and (2), respectively. Assume in addition that $S(I), P(I) \subseteq [0,1]^n \times \mathbb{R}^p$. For a vector $x \in \mathbb{R}^n$ such that $(x, y) \in P(I)$ for some $y \in \mathbb{R}^p$, let $\text{supp}(x) = \{i \in \{1, \ldots, n\} : x_i \neq 0\}$.

Fractional Decomposition Tree (FDT) is a polynomial-time algorithm that given a point $x \in P(I)$ produces a convex combination of feasible points in $S(I)$ that are dominated by a "factor" C of x. If $C = g_I$, it would be optimal. However we can only guarantee a factor of $g_I^{|\text{supp}(x)|}$. FDT relies on iteratively solving linear programs that are about the same size as the description of $P(I)$.

Theorem 2. *Assume $1 \leq g_I < \infty$. The Fractional Decomposition Tree (FDT) algorithm , given $(x^*, y^*) \in P(I)$, produces in polynomial time $\lambda \in [0,1]^k$ and $(z^1, w^1), \ldots, (z^k, w^k) \in S(I)$ such that $k \leq |\text{supp}(x^*)|$, $\sum_{i=1}^k \lambda_i z^i \leq Cx^*$, and $\sum_{i=1}^k \lambda_i = 1$. Moreover, $C \leq g_I^{|\text{supp}(x^*)|}$.*

FDT finds feasible solutions to any MILP with finite gap. This can be of independent interest, especially in proving that a model has unbounded gap.

Theorem 3. *Assume $1 \leq g_I < \infty$. The DomToIP algorithm finds $(\hat{x}, \hat{y}) \in S(I)$ in polynomial time.*

For general I it is NP-hard to even decide if $S(I)$ is empty or not. There are a number of heuristics for this purpose, such as the feasibility pump heuristic [9,10]. These heuristics are often very effective and fast in practice, however, they can sometimes fail to find a feasible solution. These heuristics do not provide any bounds on the quality of the solution they find.

We consider the following TSP-related problems. The *2-edge-connected subgraph problem (2EC)* is to find a minimum-weight 2-edge-connected (multigraph) subgraph in a graph $G = (V, E)$ with respect to weights $c \in \mathbb{R}^E_{\geq 0}$. In the *tree-augmentation problem (TAP)* we wish to add a minimum-cost set of edges to a tree to make it 2-edge-connected. We formally define TAP in Section 4.

One can extend the FDT algorithm for binary MILPs into covering $\{0,1,2\}$-MILPs by losing a factor $2^{|\operatorname{supp}(x)|}$. In order to eradicate this factor, we need to treat the coordinate i with $x_i = 1$ differently. The 2EC problem has the natural linear programming relaxation is $\mathrm{2EC}(G) = \{x \in \mathbb{R}^E_{\geq 0} \ : \ x(\delta(S)) \geq 2 \text{ for } \emptyset \subset S \subset V)\}$.

Theorem 4. *Let $G = (V, E)$ and x be an extreme point of $\mathrm{2EC}(G)$. The FDT algorithm for 2EC produces $\lambda \in [0,1]^k$ and 2-edge-connected multigraphs $F_1, \ldots, F_k$ such that $k \leq 2|V| - 1$, $\sum_{i=1}^k \lambda_i \chi^{F_i} \leq Cx^*$, and $\sum_{i=1}^k \lambda_i = 1$. Moreover, $C \leq g(\mathrm{2EC})^k$, where $g(\mathrm{2EC})$ is the integrality gap of the 2-edge-connected multigraph problem with respect to formulation in (**??**)*

Experiments Although the bound guaranteed in both Theorems 2 and 4 are very large for large problems, we show that in practice, the algorithm works very well for the TSP-like problems described above. We show how one might use FDT to investigate the integrality gap for such well-studied problems.

Known polyhedral structure makes it easier to study integrality gaps for such problems. Carr and Ravi [2] introduced fundamental extreme points. A point x in Held-Karp relaxation for TSP (or 2EC; they have the same relaxation) is a point whose support of x, namely G_x satisfies the following: i) G_x is a cubic graph, ii) in G_x there is exactly one edge with $x_e = 1$ incident to each node iii) The fractional edges of G_x form a Hamiltonian cycle. We say a fundamental extreme point (FEP) is *order k* if there are k nodes on this Hamiltonian cycle. An FEP of order k could represent an instance with many more than k vertices. Carr and co-authors [3,2,1] proved that showing that Cx is a convex combination of tours (resp. 2-edge-connected subgraphs) for all fundamental extreme points is equivalent to proving that the integrality gap for TSP (resp. 2EC) is bounded above by C. Thus we can create the “hardest” LP solutions to decompose.

There are fairly good bounds for the integrality gap for TSP or 2EC. Benoit and Boyd [7] used a quadratic program to show the integrality gap for TSP is at most $\frac{20}{17}$ for graphs with at most 10 vertices. Alexander et. al [6] used the same ideas to provide an upper bound of $\frac{7}{6}$ for 2EC on graphs with at most 10 vertices. For 2EC we show that the integrality gap is at most $\frac{6}{5}$ for FEPs of order at most 12. An FEP of order k might correspond to an extreme point of a much bigger graph, since each edge in a FEP with value 1 actually corresponds to a path of edges with value 1. For TAP, we create random fractional extreme points and round them using FDT. For the instances that we create the blow-up factor is always below $\frac{3}{2}$ providing an upper bound for such instances.

Contributions The paper has the following contributions:

– We give a simple algorithm, Dom2IP, that can prove a formulation's integrality gap is unbounded or if not provide a feasible integer solution. Someone

formulating a first MILP for a new problem can test it with Dom2IP. If the algorithm ever fails in finding a feasible solution, the MILP has an unbounded gap.
- We give an algorithm, Fractional Decomposition Tree (FDT), to construct the convex decomposition in the Carr-Vempala theorem, perhaps scaling by a factor larger than the integrality gap. Each step of this algorithm provides a *lower bound* on the instances integrality gap. This also provides a lower bound on the approximation factor of any LP-based approximation algorithm using this formulation. The overall approximation factor of the FDT algorithm is an upper bound on the integrality gap for that specific instance.
- For a special set of problems related to TSP, where there is a notion of a fundamental extreme point and long-running attempts to exactly determine the integrality gap of classic formulations, experimental analysis with FDT can help give some intuition about which bound(s) is/are likely to be loose. Computing on fundamental extreme points is a way to experimentally characterize the gap upper bound. There is no guarantee. Still, this can help direct theoretical analysis in the most promising direction. Furthermore, for these kinds of problems, especially 2EC, FDT gives good approximate solutions, better than the best current competitor (Christofides).

2 Finding a feasible solution

Consider a formulation instance $I = (A, G, b)$. Define sets $S(I)$ and $P(I)$ as in (1) and (2), respectively. And assume $S(I), P(I) \in [0,1]^n \times \mathbb{R}^p$. For simplicity in the notation we denote $P(I), S(I)$, and $g(I)$ with P, S, and g for this section and the next section. Also, for both sections we assume $t = |\operatorname{supp}(x)|$. Without loss of generality we can assume $x_i = 0$ for $i = t+1, \ldots, n$.

In this section we prove Theorem 3. In fact, we prove a stronger result.

Lemma 1. *Given $(x, y) \in \mathcal{D}(P)$, there is an algorithm (the Dom2IP algorithm) that finds $(x^{(t)}, y^{(t)}) \in S$ in polynomial time, such that $x^t \leq x$, where $t = |\operatorname{supp}(x)|$*

We prove Lemma 1 by introducing an algorithm that "fixes" the variables iteratively, starting from x_1 and ending at x_t. Suppose we run the algorithm for $\ell \in \{0, \ldots, t-1\}$ iterations and we have $(x^{(\ell)}, y^{(\ell)}) \in \mathcal{D}(P)$ such that $x_i^{(\ell)} \in \{0,1\}$ for $i = 1, \ldots, t$. Now consider the following linear program. The variables of this linear program are the $z \in \mathbb{R}^n$ variables and $w \in \mathbb{R}^p$.

$$
\begin{aligned}
\text{DominantToFeasbile}(x^{(\ell)}) \qquad & \min && z_{\ell+1} && && (4)\\
& \text{s.t.} && Az + Gw \geq b && && (5)\\
& && z_j = x_j^{(\ell)} && j = 1, \ldots, \ell && (6)\\
& && z_j \leq x_j^{(\ell)} && j = \ell+1, \ldots, n && (7)\\
& && z \geq 0 && && (8)
\end{aligned}
$$

If the optimal value to DominantToFeasbile($x^{(\ell)}$) is 0, then let $x_{\ell+1}^{(\ell+1)} = 0$. Otherwise if the optimal value is strictly positive let $x_{\ell+1}^{(\ell+1)} = 1$. Let $x_j^{(\ell+1)} = x_j^{(\ell)}$ for $j \in \{1,\ldots,n\} \setminus \{\ell+1\}$.

The above procedure suggests how to find $(x^{(\ell+1)}, y^{(\ell+1)})$ from $(x^{(\ell)}, y^{(\ell)})$. The Dom2IP algorithm initializes with $(x^{(0)}, y^{(0)}) = (x, y)$ and iteratively calls this procedure in order to obtain $(x^{(t)}, y^{(t)})$. We prove that indeed $(x^{(t)}, y^{(t)}) \in S$.

First, we need to show that in any iteration $\ell = 0,\ldots,t-1$ of DomtoIP the DominantToFeasbile(x^{ℓ}) is feasible. We show something stronger. For $\ell = 0,\ldots,t-1$ let

$$\begin{aligned} \mathrm{LP}^{(\ell)} &= \{(z,w) \in P \;:\; z \le x^{(\ell)} \text{ and } z_j = x_j^{(\ell)} \text{ for } j = 1,\ldots,\ell\}, \text{ and} \\ \mathrm{IP}^{(\ell)} &= \{(z,w) \in \mathrm{LP}^{(\ell)} \;:\; z \in \{0,1\}^n\}. \end{aligned}$$

Notice that if $\mathrm{LP}^{(\ell)}$ is a non-empty set then DominantToFeasbile($x^{(\ell)}$) is feasible. We show by induction on ℓ that $\mathrm{LP}^{(\ell)}$ and $\mathrm{IP}^{(\ell)}$ are not empty sets for $\ell = 0,\ldots,t-1$. First notice that $\mathrm{LP}^{(0)}$ is clearly feasible since by definition $(x^{(0)}, y^{(0)}) \in \mathcal{D}(P)$, meaning there exists $(z,w) \in P$ such that $z \le x^{(0)}$. By Theorem 1, there exists $(\tilde{z}^i, \tilde{w}^i) \in S$ and $\theta_i \ge 0$ for $i = 1,\ldots,k$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \le gz$. Hence, $\sum_{i=1}^k \theta_i \tilde{z}^i \le gz \le gx^{(0)}$. So if $x_j^{(0)} = 0$, then $\sum_{i=1}^k \theta_i \tilde{z}_j^i = 0$, which implies that $\tilde{z}_j^i = 0$ for all $i = 1,\ldots,k$ and $j = 1,\ldots,n$ where $x_j^{(0)} = 0$. Hence, $z^i \le x^{(0)}$ for $i = 1,\ldots,k$. Therefore $(\tilde{z}^i, \tilde{w}^i) \in \mathrm{IP}^{(0)}$ for $i = 1,\ldots,k$, which implies $\mathrm{IP}^{(0)} \neq \emptyset$.

Now assume $\mathrm{IP}^{(\ell)}$ is non-empty for some $\ell \in \{0,\ldots,t-2\}$. Since $\mathrm{IP}^{(\ell)} \subseteq \mathrm{LP}^{(\ell)}$ we have $\mathrm{LP}^{(\ell)} \neq \emptyset$ and hence the DominantToFeasbile($x^{(\ell)}$) has an optimal solution (z^*, w^*).

We consider two cases. In the first case, we have $z_{\ell+1}^* = 0$. In this case we have $x_{\ell+1}^{(\ell+1)} = 0$. Since $z^* \le x^{(\ell+1)}$, we have $(z^*, w^*) \in \mathrm{LP}^{(\ell+1)}$. Also, $(z^*, w^*) \in P$. By Theorem 1 there exists $(\tilde{z}^i, \tilde{w}^i) \in S$ and $\theta_i \ge 0$ for $i = 1,\ldots,k$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \le gz^*$. We have $\sum_{i=1}^k \theta_i \tilde{z}^i \le gz^* \le gx^{(\ell+1)}$ So for $j \in \{1,\ldots,n\}$ where $x_j^{(\ell+1)} = 0$, we have $z_j^i = 0$ for $i = 1,\ldots,k$. Hence, $\tilde{z}^i \le x^{(\ell+1)}$ for $i = 1,\ldots,k$. Hence, there exists $(z,w) \in S$ such that $z \le x^{(\ell+1)}$. We claim that $(z,w) \in \mathrm{IP}^{(\ell+1)}$. If $(z,w) \notin \mathrm{IP}^{(\ell+1)}$ we must have $1 \le j \le \ell$ such that $z_j < x_j^{(\ell+1)}$, and thus $z_j = 0$ and $x_j^{(\ell+1)} = 1$. Without loss of generality assume j is minimum number satisfying $z_j < x_j^{(\ell+1)}$. Consider iteration j of the Dom2IP algorithm. Notice that $z \le x^{(\ell+1)} \le x^{(j)}$. We have $x_j^{(j)} = 1$ which implies when we solved DominantToFeasbile($x^{(j-1)}$) the optimal value was strictly larger than zero. However, (z,w) is a feasible solution to DominantToFeasbile($x^{(j-1)}$) and gives an objective value of 0. This is a contradiction, so $(z,w) \in \mathrm{IP}^{(\ell+1)}$.

Now for the second case, assume $z_{\ell+1}^* > 0$. We have $x_{\ell+1}^{(\ell+1)} = 1$. Notice that for each point $z \in \mathrm{LP}^{(\ell)}$ we have $z_{\ell+1} > 0$, so for each $z \in \mathrm{IP}^{(\ell)}$ we have $z_{\ell+1} > 0$, i.e. $z_{\ell+1} = 1$. This means that $(z,w) \in \mathrm{IP}^{(\ell+1)}$, and $\mathrm{IP}^{(\ell+1)} \neq \emptyset$.

Now consider $(x^{(t)}, y^{(t)})$. Let $(z, y^{(t)})$ be the optimal solution to $\mathrm{LP}^{(t-1)}$. If $x^{(t)} = 0$, we have $x^{(t)} = z$, which implies that $(x^{(t)}, y^{(t)}) \in P$, and since $x^{(t)} \in \{0,1\}^n$ we have $(x^{(t)}, y^{(t)}) \in S$. If $x^{(t)} = 1$, it must be the case that $z_t > 0$. By the argument above there is a point $(z', w') \in \mathrm{IP}^{(t-1)}$. We show that $x^{(t)} = z'$. For $j = 1, \ldots, n-1$ we have $z'_j = x_j^{(t-1)} = x_j^{(t)}$. We just need to show that $z'_j = 1$. Assume $z'_j = 0$ for contradiction, then $(z', w') \in \mathrm{LP}^{(t-1)}$ has objective value of 0 for DominantToFeasbile$(x^{(t-1)})$, this is a contradiction to (z, w) being the optimal solution. This concludes the proof of Lemma 1.

Notice that Lemma 1 implies Theorem 3, since it is easy to obtain an integer point in $\mathcal{D}(P)$: rounding up any fractional point in P gives us a point in $\mathcal{D}(P)$.

3 FDT on binary MIPs

Assume we are given a point $(x^*, y^*) \in P$. For instance, (x^*, y^*) can be the optimal solution of minimizing a cost function cx over set P, which provides a lower bound on $\min_{(x,y)\in S(I)} cx$. In this section, we prove Theorem 2 by describing the Fractional Decomposition Tree (FDT) algorithm.

The FDT algorithm grows a tree similar to the classic branch-and-bound search tree for integer programs. Each node represents a partially integral vector $(\bar{x}, \bar{y})$ in $\mathcal{D}(P)$ together with a multiplier $\bar{\lambda}$. The solutions contained in the nodes of the tree become progressively more integral at each level. In each level of the tree, the algorithm maintain a conic combination of points with the properties mentioned above. Leaves of the FDT tree contain solutions with integer values for all the x variables that dominate a point in P. We will later see how we can turn these into points in S.

Branching on a node We begin with the following lemmas that show how the FDT algorithm branches on a variable.

Lemma 2. *Given $(x', y') \in \mathcal{D}(P)$ and $\ell \in \{1, \ldots, n\}$, we can find in polynomial time vectors $(\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ and scalars $\gamma_0, \gamma_1 \in [0,1]$ such that: (i) $\gamma_0 + \gamma_1 \geq \frac{1}{g}$, (ii) $(\hat{x}^0, \hat{y}^0)$ and $(\hat{x}^1, \hat{y}^1)$ are in P ,(iii) $\hat{x}^0_\ell = 0$ and $\hat{x}^1_\ell = 1$, (iv) $\gamma_0 \hat{x}^0 + \gamma_1 \hat{x}^1 \leq x'$.*

Proof. Consider the following linear program which we denote by Branching(ℓ, x', y'). The variables of Branching(ℓ, x', y') are γ_0, γ_1 and (x^0, y^0) and (x^1, y^1).

$$
\begin{array}{llll}
\text{Branching}(\ell, x', y') & \max & \lambda_0 + \lambda_1 & (9)\\
& \text{s.t.} & Ax^j + Gy^j \geq b\lambda_j \quad \text{for } j = 0,1 & (10)\\
& & 0 \leq x^j \leq \lambda_j \quad \text{for } j = 0,1 & (11)\\
& & x^0_\ell = 0,\ x^1_\ell = \lambda_1 & (12)\\
& & x^0 + x^1 \leq x' & (13)\\
& & \lambda_0, \lambda_1 \geq 0 & (14)
\end{array}
$$

Let $(x^0, y^0), (x^1, y^1)$, and γ_0, γ_1 be an optimal solution solution to the LP above. Let $(\hat{x}^0, \hat{y}^0) = (\frac{x^0}{\gamma_0}, \frac{y^0}{\gamma_0})$, $(\hat{x}^1, \hat{y}^1) = (\frac{x^1}{\gamma_1}, \frac{y^1}{\gamma_1})$. This choice satisfies (ii), (iii), (iv). To show that (i) is also satisfied we prove the following claim.

Claim. We have $\gamma_0 + \gamma_1 \geq \frac{1}{g}$.

Proof. We show that there is a feasible solution that achieves the objective value of $\frac{1}{g}$. By Theorem 1 there exists $\theta \in [0,1]^k$, with $\sum_{i=1}^k \theta_i = 1$ and $(\tilde{x}^i, \tilde{y}^i) \in S$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq g x'$.

$$x' \geq \sum_{i=1}^{k} \frac{\theta_i}{g} \tilde{x}^i = \sum_{i \in [k]: \tilde{x}^i_\ell = 0} \frac{\theta_i}{g} \tilde{x}^i + \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g} \tilde{x}^i \tag{15}$$

For $j = 0, 1$, let $(x^j, y^j) = \sum_{i \in [k]: \tilde{x}^i_\ell = j} \frac{\theta_i}{g} (\tilde{x}^i, \tilde{y}^i)$. Also let $\lambda_0 = \sum_{i \in [k]: \tilde{x}^i_\ell = 0} \frac{\theta_i}{g}$ and $\lambda_1 = \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g}$. Note that $\lambda_0 + \lambda_1 = \frac{1}{g}$. Constraint (13) is satisfied by Inequality (15). Also, for $j = 0, 1$ we have

$$Ax^j + Gy^j = \sum_{i \in [k], \tilde{x}^i_\ell = j} \frac{\theta_i}{g} (A\tilde{x}^i + G\tilde{y}^i) \geq b \sum_{i \in [k], \tilde{x}^i_\ell = j} \frac{\theta_i}{g} = b\lambda_j. \tag{16}$$

Hence, Constraints (10) holds. Constraint (12) also holds since x^0_ℓ is obviously 0 and $x^1_\ell = \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g} = \lambda_1$. The rest of the constraints trivially hold. □

This concludes the proof of Lemma 2. □

We now show if x' in the statement of Lemma 2 is partially integral, we can find solutions with more integral components.

Lemma 3. *Given $(x', y') \in \mathcal{D}(P)$, such that $x'_1, \ldots, x'_{\ell-1} \in \{0, 1\}$ for some $\ell \geq 1$, we can find in polynomial time vectors $(\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ and scalars $\gamma_0, \gamma_1 \in [0, 1]$ such that: (i) $\frac{1}{g} \leq \gamma_0 + \gamma_1 \leq 1$, (ii) $(\hat{x}^0, \hat{y}^0)$ and $(\hat{x}^1, \hat{y}^1)$ are in $\mathcal{D}(P)$, (iii) $\hat{x}^0_\ell = 0$ and $\hat{x}^1_\ell = 1$, (iv) $\gamma_0 \hat{x}^0 + \gamma_1 \hat{x}^1 \leq x'$,(v) $\hat{x}^i_j \in \{0, 1\}$ for $i = 0, 1$ and $j = 1, \ldots, \ell - 1$.*

Proof. By Lemma 2 we can find $(\bar{x}^0, \bar{y}^0)$, $(\bar{x}^1, \bar{y}^1)$, γ_0 and γ_1 that satisfy (i), (ii), (iii), and (iv). We define $\hat{x}^0$ and $\hat{x}^1$ as follows. For $i = 0, 1$, for $j = 1, \ldots, \ell - 1$, let $\hat{x}^i_j = \lceil \bar{x}^i_j \rceil$, for $j = \ell, \ldots, t$ let $\hat{x}^i_j = \bar{x}^i_j$. We now show that $(\hat{x}^0, \bar{y}^0)$, $(\hat{x}^1, \bar{y}^1)$, γ_0, and γ_1 satisfy all the conditions. Note that conditions (i), (ii), (iii), and (v) are trivially satisfied. Thus we only need to show (iv) holds. We need to show that $\gamma_0 \hat{x}^0_j + \gamma_1 \hat{x}^1_j \leq g x'_j$. If $j = \ell, \ldots, t$, then this clearly holds. Hence, assume $j \leq \ell - 1$. By the property of x' we have $x'_j \in \{0, 1\}$. If $x'_j = 0$, then by Constraint (13) we have $\bar{x}^0_j = \bar{x}^1_j = 0$. Therefore, $\hat{x}^i_j = 0$ for $i = 0, 1$, so (iv) holds. Otherwise if $x'_j = 1$, then we have $\gamma_0 \hat{x}^0_j + \gamma_1 \hat{x}^1_j \leq \gamma_0 + \gamma_1 \leq 1 \leq x'_j$. Therefore (v) holds. □

Growing and Pruning FDT tree The FDT algorithm maintains nodes L_i in iteration i of the algorithm. The nodes in L_i correspond to the nodes in level L_i of the FDT tree. The points in the leaves of the FDT tree, L_t, are points in $\mathcal{D}(P)$ and are integral for all integer variables.

Lemma 4. *There is a polynomial time algorithm that produces sets $L_0, \dots, L_t$ of pairs of $(x, y) \in \mathcal{D}(P)$ together with multipliers λ with the following properties for $i = 0, \dots, t$: (a) If $(x, y) \in L_i$, then $x_j \in \{0, 1\}$ for $j = 1, \dots, i$, i.e. the first i coordinates of a solution in level i are integral, (b) $\sum_{[(x,y),\lambda] \in L_i} \lambda \geq \frac{1}{g^i}$, (c) $\sum_{[(x,y),\lambda] \in L_i} \lambda x \leq x^*$, (d) $|L_i| \leq t$.*

Proof. We prove this lemma using induction but one can clearly see how to turn this proof into a polynomial time algorithm. Let L_0 be the set that contains a single node (*root of the FDT tree*) with (x^*, y^*) and multiplier 1. It is easy to check all the requirements in the lemma are satisfied for this choice.

Suppose by induction that we have constructed sets $L_0, \dots, L_i$. Let the solutions in L_i be (x^j, y^j) for $j = 1, \dots, k$ and λ_j be their multipliers, respectively. For each $j = 1, \dots, k$ by Lemma 3 (setting $(x', y') = (x^j, y^j)$ and $\ell = i + 1$) we can find $(x^{j0}, y^{j0}), (x^{j1}, y^{j1})$ and λ_j^0, λ_j^1 with the properties (i) to (v) in Lemma 3. Define L' to be the set of nodes with solutions $(x^{j0}, y^{j0}), (x^{j1}, y^{j1})$ and multipliers $\lambda_j \lambda_j^0$, $\lambda_j \lambda_j^1$, respectively, for $j = 1, \dots, k$. It is easy to check that set L' is a suitable candidate for L_{i+1}, i.e. set L' satisfies (a), (b) and (c). However we can only ensure that $|L'| \leq 2k \leq 2t$, and might have $|L'| > t$. We call the following linear program Pruning(L'). Let $L' = \{[(x^1, y^1), \gamma_1], \dots, [(x^{2k}, y^{2k}), \gamma_{2k}]\}$. The variables of Pruning(L') is a scalar variable θ_j for each node j in L'.

$$\text{Pruning}(L') \qquad \max \quad \sum_{j=1}^{2k} \theta_j \tag{17}$$

$$\text{s.t.} \quad \sum_{j=1}^{2k} \theta_j x_i^j \leq x_i^* \qquad \text{for } i = 1, \dots, t \tag{18}$$

$$\theta \geq 0 \tag{19}$$

Notice that $\theta = \gamma$ is in fact a feasible solution to Pruning(L'). Let θ^* be the optimal vertex point solution to this LP. Since the problem is in $\mathbb{R}^{2k}$, θ^* has to satisfy $2k$ linearly independent constraints at equality. However, there are only t constraints of type (18). Therefore, there are at most t coordinates of θ_j^* that are non-zero. We claim that L_{i+1} which consists of (x^j, y^j) for $j = 1, \dots, 2k$ and their corresponding multipliers θ_j^* satisfy the properties in the statement of the lemma. Notice that, we can discard the nodes in L_{i+1} that have $\theta_j^* = 0$, so $|L_{i+1}| \leq t$. Also, since θ^* is optimal and γ is feasible for Pruning(L'), we have $\sum_{j=1}^{k} \theta_j^* \geq \sum_{j=1}^{2k} \gamma_j \geq \frac{1}{g^{i+1}}$. □

From leaves of FDT to feasible solutions For the leaves of the FDT tree, L_t, we have that every solution (x, y) in L_t has $x \in \{0, 1\}^n$ and $(x, y) \in \mathcal{D}(P)$. By applying Lemma 1 we can obtain a point $(x', y') \in S$ such that $x' \leq x$. This conclude the description of the FDT algorithm and proves Theorem 2.

4 Computational experiments with FDT

We ran FDT on two covering problem: tree augmentation (TAP) and 2EC. For TAP we are given a tree $T = (V, E)$, and a set of non-edges L between vertices

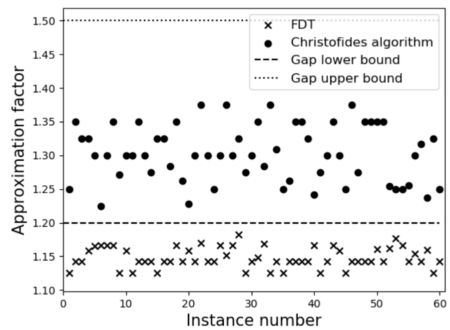

Fig. 1. Christofides' algorithm vs FDT on all fundamental extreme points of order 10.

in V and costs $c \in \mathbb{R}^L_{\geq 0}$. We wish to find the minimum-cost $L' \subseteq L$ such that $T+L'$ is 2-edge-connected. For $\ell \in L$, let P_ℓ be the set of edges in the unique path between the endpoints of ℓ in T. For TAP, $S(\text{TAP}) = \{x \in \mathbb{Z}^L_{\geq 0} : \sum_{\ell : e \in P_\ell} x_\ell \geq 1,\ \text{for } e \in E\}$ and $P(\text{TAP})$ relaxes integrality. We know $\frac{3}{2} \leq g(\text{TAP}) \leq 2$ [4,5]. We ran binary FDT on a set of 264 fractional extreme points of $P(\text{TAP})$. Table **??** shows FDT found solutions better than the integrality-gap lower bound for most instances.

We also implemented the polyhedral version of Christofides' algorithm [19]. Figure 1 shows FDT's solutions on fundamental extreme points of order 10 are always better than those from Christofides' algorithm. We ran FDT for 2EC

	$C \in [1.1, 1.2]$	$C \in (1.2, 1.3]$	$C \in (1.3, 1.4]$	$C \in (1.4, 1.5]$
TAP	36	66	170	10

Table 1. The scale factor C for FDT run on 264 randomly generated TAP instances with fractional extreme points: 138 instances have 74 variables. The rest have 250.

on 963 fractional extreme points of $2\text{EC}(G)$. We enumerated all fundamental vertices of order 10 and 12. Table 2 shows that again FDT found solutions better than the integrality-gap lower bound for most instances.

	$C \in [1.08, 1.11]$	$C \in (1.11, 1.14]$	$C \in (1.14, 1.17]$	$C \in (1.17, 1.2]$
2EC	79	201	605	78

Table 2. FDT for 2EC implemented applied to all fundamental extreme points of order 10 or 12. A FEP of order k has $\frac{3k}{2}$ variables. The lower bound on $g(2\text{EC})$ is $\frac{6}{5}$.

References

1. Sylvia Boyd and Robert Carr. Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. *Discrete Optimization*, 8(4):525–539, 2011.
2. Robert Carr and R. Ravi. A new bound for the 2-edge connected subgraph problem. In *Proceedings of the 6th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 112–125, London, UK, UK, 1998. Springer-Verlag.
3. Robert Carr and Santosh Vempala. On the Held-Karp relaxation for the asymmetric and symmetric traveling salesman problems. *Mathematical Programming*, 100(3):569–587, Jul 2004.
4. Greg N. Frederickson and Joseph Ja'Ja'. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981.
5. Cheriyan, Joseph and Karloff, Howard and Khandekar, Rohit and Könemann, Jochen. On the integrality ratio for tree augmentation. *Operations Research Letters*, 36(4):399–401, 2008.
6. Anthony Alexander, Sylvia Boyd, and Paul Elliott-Magwood. On the Integrality Gap of the 2-Edge Connected Subgraph Problem. Technical report, University of Ottawa, Ottawa, Canada, 04 2006.
7. Geneviève Benoit and Sylvia Boyd. Finding the exact integrality gap for small traveling salesman problems. *Mathematics of Operations Research*, 33(4):921–931, 2008.
8. Gérard Cornuéjols, Jean Fonlupt, and Denis Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33(1):1–27, Sep 1985.
9. Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, Sep 2005.
10. Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1(2):201–222, Oct 2009.
11. M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.
12. Michel X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Math. Program.*, 69(2):335–349, August 1995.
13. Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Second corrected edition edition, 1993.
14. Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, FOCS '98, pages 448–, Washington, DC, USA, 1998. IEEE Computer Society.
15. Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, Dec 1987.
16. A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.
17. Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.
18. David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.
19. Laurence A. Wolsey. *Heuristic analysis, linear programming and branch and bound*, pages 121–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 1980.

A FDT for 2EC

In Section 3 our focus was on binary MIPs. In this section, in an attempt to extend FDT to 0,1,2 problems we introduce an FDT algorithm for a 2-edge-connected multigraph problem. Given a graph $G = (V, E)$ a multi-subset of edges F of G is a 2-edge-connected multigraph of G if for each set $\emptyset \subset U \subset V$, the number of edge in F that have one endpoint in U and one not in U is at least 2. In the 2EC problem, we are given non-negative costs on the edge of G and the goal is to find the minimum cost 2-edge-connected multigraph of G. Notice that, no optimal solution ever takes 3 copies of an edge in 2EC, hence we assume that we can take an edge at most 2 times. The natural linear programming relaxation is $\text{2EC}(G) = \{x \in [0,2]^E : x(\delta(U)) \geq 2 \text{ for } \emptyset \subset U \subset V\}$. Notice that $\mathcal{D}(\text{2EC}(G)) \cap [0,2]^E = \text{2EC}(G)$, since 2EC is a covering problem. We want to prove Theorem 4. We do not know the exact value for g_{2EC}, but we know $\frac{6}{5} \leq g_{\text{2EC}} \leq \frac{3}{2}$ [2,19]. Also, we need to remark that polyhedral version of Christofides' algorithm provides a $\frac{3}{2}$-approximation for 2EC, i.e. we already have an algorithm with $C \leq \frac{3}{2}$. However, we showed in Section 4 that in practice the constant C for the FDT algorithm for 2EC is much better than $\frac{3}{2}$ for fundamental extreme points with 10 vertices.

The FDT algorithm for 2EC is very similar to the one for binary MILPs, but there are some differences as well. A natural thing to do is to have three branches for each node of the FDT tree, however, the branches that are equivalent to setting a variable to 1, might need further decomposition. That is the main difficulty when dealing with $\{0,1,2\}$-MILPs.

First, we need a branching lemma. Observe that the following branching lemma is essentially a translation of Lemma 2 for 0,1,2 problems except for one additional clause.

Lemma 5. *Given $x \in \text{2EC}(G)$, and $e \in E$ we can find in polynomial time vectors x^0, x^1 and x^2 and scalars γ_0, γ_1, and γ_2 such that: (i) $\gamma_0+\gamma_1+\gamma_2 \geq \frac{1}{g_{\text{2EC}}}$, (ii) x^0, x^1, and x^2 are in $\text{2EC}(G)$, (iii) $x^0_e = 0$, $x^1_e = 1$, and $x^2_e = 2$, (iv) $\gamma_0 x^0 + \gamma_1 x^1 + \gamma_2 x^2 \leq x$, (v) for $f \in E$ with $x_f \geq 1$, we have $x^j_f \geq 1$ for $j = 0, 1, 2$.*

Proof. Consider the following LP with variables λ_j and x^j for $j = 0, 1, 2$.

$$\max \quad \sum_{j=0,1,2} \lambda_j \tag{20}$$

$$\text{s.t.} \quad x^j(\delta(U)) \geq 2\lambda_j \qquad \text{for } \emptyset \subset U \subset V, \text{ and } j = 0,1,2 \tag{21}$$

$$0 \leq x^j \leq 2\lambda_j \qquad \text{for } j = 0,1,2 \tag{22}$$

$$x^j_e = j \qquad \text{for } j = 0,1,2 \tag{23}$$

$$x^j_f \geq j \qquad \text{for } f \in E \text{ where } x_f \geq 1, \text{ and } j = 0,1,2 \tag{24}$$

$$x^0 + x^1 + x^2 \leq x \tag{25}$$

$$\lambda_0, \lambda_1, \lambda_2 \geq 0 \tag{26}$$

Let x^j, γ_j for $j = 0, 1, 2$ be an optimal solution solution to the LP above. Let $\hat{x}^j = \frac{x^j}{\gamma_j}$ for $j = 0, 1, 2$ where $\gamma_j > 0$. If $\gamma_j = 0$, let $\hat{x}^j = 0$. Observe that (ii), (iii), (iv), and (v) are satisfied with this choice. We can also show that $\gamma_0 + \gamma_1 + \gamma_2 \geq \frac{1}{g_{2\mathrm{EC}}}$, which means that (i) is also satisfied. The proof is similar to the proof of the claim in Lemma 2, but we need to replace each $f \in E$ with $x_f \geq 1$ with a suitably long path to ensure that Constraint (24) is also satisfied. We skip the details. □

In contrast to FDT for binary MIPs where we round up the fractional variables that are already branched on at each level, in FDT for 2EC we keep all coordinates as they are and perform a rounding procedure at the end. Formally, let L_i for $i = 1, \ldots, |\operatorname{supp}(x^*)|$ be collections of pairs of feasible points in $2\mathrm{EC}(G)$ together with their multipliers. Let $t = |\operatorname{supp}(x^*)|$ and assume without loss of generality that $\operatorname{supp}(x^*) = \{e_1, \ldots, e_t\}$.

Lemma 6. *The FDT algorithm for 2EC in polynomial time produces sets $L_0, \ldots, L_t$ of pairs $x \in 2\mathrm{EC}(G)$ together with multipliers λ with the following properties. (a) If $x \in L_i$, then $x_{e_j} = 0$ or $x_{e_j} \geq 1$ for $j = 1, \ldots, i$, (b) $\sum_{(x,\lambda)\in L_i} \lambda \geq \frac{1}{g^i_{2\mathrm{EC}}}$, (c) $\sum_{(x,\lambda)\in L_i} \lambda x \leq x^*$, (d) $|L_i| \leq t$.*

The proof is similar to Lemma 4, but we need to use property (i) in Lemma 5 to prove that (a) also holds.

Consider the solutions x in L_t. For each variable e we have $x_e = 0$ or $x_e \geq 1$.

Lemma 7. *Let x be a solution in L_t. Then $\lfloor x \rfloor \in 2\mathrm{EC}(G)$.*

Proof. Suppose not. Then there is a set of vertices $\emptyset \subset U \subset V$ such that $\sum_{e\in\delta(U)} \lfloor x_e \rfloor < 2$. Since $x \in 2\mathrm{EC}(G)$ we have $\sum_{e\in\delta(U)} x_e \geq 2$. Therefore, there is an edge $f \in \delta(U)$ such that x_f is fractional. By property (a) in Lemma 6, we have $1 < x_f < 2$. Therefore, there is another edge h in $\delta(U)$ such that $x_h > 0$, which implies that $x_h \geq 1$. But in this case $\sum_{e\in\delta(U)} \lfloor x_e \rfloor \geq \lfloor x_f \rfloor + \lfloor x_h \rfloor \geq 2$. This is a contradiction. □

The FDT algorithm for 2EC iteratively applies Lemmas 5 and 6 to variables $x_1, \ldots, x_t$ to obtain leaf point solutions L_t. Then, we just need to apply Lemma 7 to obtain the 2-edge-connected multigraphs from every solution in L_t. Notice that since x is an extreme point we have $t \leq 2|V| - 1$ [8]. By Lemma 6 we have

$$\sum_{(x,\lambda)\in L_t} \frac{\lambda}{\sum_{(x,\lambda)\in L_t} \lambda} \lfloor x \rfloor \leq \frac{1}{\sum_{(x,\lambda)\in L_t} \lambda} \sum_{(x,\lambda)\in L_t} \lambda x \leq g^t_{2\mathrm{EC}} x^*.$$