Fractional Decomposition Trees: A tool for studying Mixed-Integer Program integrality gaps

Robert Carr* Arash Haddadan† Cynthia A. Phillips‡

May 13, 2019

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 $S(A, G, b)$,

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

*University of New Mexico, bobcarr@unm.edu

†Carnegie Mellon University, ahaddada@andrew.cmu.edu

‡Sandia National Laboratories caphill@sandia.gov

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 Mixed-Integer-Linear Program (MILP) is min $\{cx : (x, y) \in S(I)\}$. It is NP-hard even to determine if an 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 an 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 [15, 16]) 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 on a parameter of the MILP, 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 $\frac{3}{2}$ [17] and at least $\frac{6}{5}$ [4]. 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 $P(I) = \text{conv}(S(I))$, 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 [14] 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 [5] 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 [5])**.** *Let $(x, y) \in P(I)$, there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $(\tilde{x}^i, \tilde{y}^i) \in S(I)$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq Cx$ if and only if $g_I \leq C$.*

While there is an exact algorithm for problems with gap 1, Theorem 1 is does not suggest an efficient construction. To study integrality gaps, we wish to find such a solution efficiently.

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 $x \in \mathbb{R}^n_{\geq 0}$ 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 $(x, y) \in P(I)$ produces a convex combination of feasible points in $S(I)$ that are dominated by a "factor" C of x in the coordinates corresponding to 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 [8, 9]. 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 which can contain multiple copies of each edge) 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 5.

One can extend the FDT algorithm for binary MILPs into covering $\{0, 1, 2\}$-MILPs by losing a factor $2^{|\operatorname{supp}(x)|}$ on top of the loss for FDT. In order to eradicate this extra factor, we need to treat the coordinate i with $x_i = 1$ differently. For 2EC we are able to achieve this. The 2EC problem has the natural linear programming relaxation is

$$\text{2EC}(G) = \{x \in [0, 2]^E \ : \ x(\delta(U)) \geq 2 \text{ for } \emptyset \subset U \subset V)\}. \tag{4}$$

We prove the following theorem.

Theorem 4. *Let $G = (V, E)$ and x be an extreme point of* 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, $C \leq g^k_{\text{2EC}}$, where g_{2EC} is the integrality gap of the 2-edge-connected multigraph problem with respect to the formulation in (4).*

Our final result is a stronger characterization of integrality gap than that in Theorem 1 for bounded covering problems. Consider an instance I where $p = 0$ (pure integer program case). By Theorem 1, integrality gap g_I is the smallest number C_1 such that for each $x \in P(I)$, there is convex combination of points $z^1, \ldots, z^k$ in $S(I)$ that is dominated by $C_1 x$, i.e. $\sum_{i=1}^k \lambda_i z^i \leq C_1 x$. Note this clearly implies $\sum_{i=1}^k \lambda_i z^i_e \leq C_1 x_e$. This allows us to introduce a lower bound on the integrality gap: let C_2 be the smallest number such that for each $x \in P(I)$, there is convex combination of points in $S(I)$ such that for all $x \in P(I)$ there is a convex combination $\sum_{i=1}^k \lambda_i z^i$ of point in $S(I)$ such that $\sum_{i=1}^k \lambda_i z^i \leq C_2 x$. As argued above $C_2 \leq C_1$. Here, we can see that $\sum_{i=1}^k \lambda_i \langle y \cdot z^i \rangle \leq C_2 \langle y \cdot x \rangle$. In particular we have $\sum_{i=1}^k \lambda_i \langle A_j \cdot z^i \rangle \leq C_2 \langle A_j \cdot x \rangle$, where A_j is the j-th row of A for some $j = 1, \ldots, m$. Even more particularly, $\sum_{i=1}^k \lambda_i \langle A_j \cdot z^i \rangle \leq C_2 b_j$ for $j \in \{1, \ldots, m\}$ such that $A_j x = b_j$, a tight constraints of x. This motivates us to define C_3 which is the smallest number such that for all $x \in P(I)$ there is convex combination of points $z_1, \ldots, z_k$ in $S(I)$ such that $\sum_{i=1}^k \theta_i \langle A_j, z^i \rangle \leq C_3 b_j$ for

j such that $A_j x = b_j$. We have $C_3 \leq C_2 \leq C_1$. We show for bounded covering problems $C_3 = C_2 = C_1 = g_I$.

For this purpose assume $P = \{x \in \mathbb{R}^n_{\geq 0} : Ax \geq b \cdot \mathbf{1}, x \leq b \cdot \mathbf{1}\}$, where $A \in \mathbb{Z}^{m \times n}_{\geq 0}$ and $b \in \mathbb{Z}_{\geq 0}$. Here, $\mathbf{1}$ is the vector of all ones of suitable dimension. Let $S = P \cap \mathbb{Z}^n$ and $g = \max_{c \geq 0} \frac{\min_{x \in S} cx}{\min_{x \in P} cx}$. Examples of problems whose natural linear programming relaxation is P (for some matrix A and integer b) include the 2-edge-connected multigraph problem, Steiner tree problem, tree augmentation problem, and many others. We call this general problem a bounded covering problem.

Theorem 5. *Let $x \in P$, there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $\tilde{x}^i \in S$ for $i = 1, \ldots, k$ such that*

- *for $\ell \in \{1, \ldots, n\}$, if $x_\ell = 0$, then $\tilde{x}^i_\ell = 0$ for $i = 1, \ldots, k$, i.e. $\tilde{x}^i$ is in the support of x,*
- *for $j \in \{1, \ldots, m\}$ with $A_j x = b$, we have $A_j(\sum_{i=1}^k \theta_i \tilde{x}^i) \leq C \cdot A_j x$,*

if and only if $C \geq g$.

This means in order to prove an upper bound on the integrality gap of a bounded covering problem, we only need to show there is a convex combination of integer feasible points in the support of the optimal fractional solution that is "cheap" on all of its tight cuts. Notice that Theorem 1 requires the certificate convex combination to be "cheap" on every single variable.

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 [4] 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 [5, 4, 3] proved that showing that Cx is a convex combination of tours (resp. 2-edge-connected multigraphs) for all fundamental extreme points is equivalent to proving that the integrality gap for TSP (resp. 2EC) is bounded above by C. We use fundamental extreme points to create the "hardest" LP solutions to decompose.

There are fairly good bounds for the integrality gap for TSP or 2EC. Benoit and Boyd [2] 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 [1] 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, DomToIP, that can prove a binary 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 DomToIP. If the algorithm ever fails in finding a feasible solution, the MILP has an unbounded gap. A feasible solution can be used to obtain new valid cuts to be added to the MILP formulation.

- 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 integrality gap of the instances. 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. In computational pursuit of improved lower bounds for integrality gap, FDT can be used to massively prune the instances.

- 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. For instance for 2EC, FDT gives good approximate solutions, better than the best current competitor (Christofides' algorithm).

- We give a stronger characterization of integrality gap for bounded covering problems. Many important combinatorial optimization problems are bounded covering problems, for which finding improved upper bounds on the integrality gap has been open for decades.

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. Assume $S(I), P(I) \subseteq [0,1]^n \times \mathbb{R}^p$. 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 $i = t+1, \dots, n$. For polyhedron L in $\mathbb{R}^{n \times p}$ let $\mathcal{D}(L) = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : x \geq x' \text{ for some } (x', y') \in L\}$, and we call $\mathcal{D}(L)$ the dominant of L.

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

Lemma 1. *Given $(x, y) \in \mathcal{D}(P)$ where $x \in \mathbb{Z}^n$, there is an algorithm (the DomToIP algorithm) that finds $(\bar{x}, \bar{y}) \in S$ in polynomial time, such that $\bar{x} \leq x$.*

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 $\ell \in \{0, \dots, t-1\}$ iterations and we have $(x^{(\ell)}, y^{(\ell)}) \in \mathcal{D}(P)$ such that $x_i^{(\ell)} \in \{0, 1\}$ for $i = 1, \dots, n$. Now consider the following linear program. The variables of this linear program are the $z \in \mathbb{R}^n$ variables and $w \in \mathbb{R}^p$.

$$
\begin{aligned}
\text{DomToFeasible}(x^{(\ell)}) \qquad \min \quad & z_{\ell+1} && (5)\\
\text{s.t.} \quad & Az + Gw \geq b && (6)\\
& z_j = x_j^{(\ell)} \quad j = 1, \dots, \ell && (7)\\
& z_j \leq x_j^{(\ell)} \quad j = \ell+1, \dots, n && (8)\\
& z \geq 0 && (9)
\end{aligned}
$$

If the optimal value to DomToFeasible$(x^{(\ell)})$ 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 $j \in \{1, \dots, n\} \setminus \{\ell+1\}$ (See Algorithm 1).

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

Algorithm 1: The DomToIP algorithm

Input: $(x, y) \in \mathcal{D}(P)$, $x \in \mathbb{Z}^n$

Output: $(x^{(t)}, y^{(t)}) \in S$, $x^{(t)} \leq x$

```
x^(0) ← x
for ℓ = 0 to t − 1 do
    x^(ℓ+1) ← x^(ℓ)
    η ← optimal value of DomToFeasible(x^(ℓ))
    y^(ℓ+1) ← optimal solution for w variables in DomToFeasible(x^(ℓ))
    if η = 0 then
        x^(ℓ+1)_{ℓ+1} ← 0
    else
        x^(ℓ+1)_{ℓ+1} ← 1
    end
end
```

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 feasible region of DomToFeasible(x^ℓ) is non-empty. We show something stronger. For $\ell = 0, \ldots, t-1$ let

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

Notice that if $\mathrm{LP}^{(\ell)}$ is a non-empty set then DomToFeasible$(x^{(\ell)})$ 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 non-empty since by definition $(x^{(0)}, y^{(0)}) \in \mathcal{D}(P)$, meaning there exists $(z, w) \in P$ such that $z \leq x^{(0)}$. By Theorem 1, there exists $(\tilde{z}^i, \tilde{w}^i) \in S$ and $\theta_i \geq 0$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz$. Thus we have $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz \leq gx^{(0)}$. 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 $x_j^{(0)} = 0$. Also recall that $x^{(0)} \in \mathbb{Z}^n$. Hence, $\tilde{z}^i \leq x^{(0)}$ 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 DomToFeasible$(x^{(\ell)})$ 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 $z^* \leq x^{(\ell+1)}$, 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 $\theta_i \geq 0$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz^*$. We have $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz^* \leq gx^{(\ell+1)}$ 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, $\tilde{z}^i \leq x^{(\ell+1)}$ for $i = 1, \ldots, k$. Hence, there exists $(z, w) \in S$ such that $z \leq x^{(\ell+1)}$. We claim that $(z, w) \in \mathrm{IP}^{(\ell+1)}$. If $(z, w) \notin \mathrm{IP}^{(\ell+1)}$ we must have $1 \leq j \leq \ell$ 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 DomToIP algorithm. Notice that $z \leq x^{(\ell+1)} \leq x^{(j-1)} \leq x^{(j)}$. We have $x_j^{(j)} = 1$ which implies when we solved DomToFeasible($x^{(j-1)}$) the optimal value was strictly larger than zero. However, (z, w) is a feasible solution to DomToFeasible($x^{(j-1)}$) 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 DomToFeasible($x^{(t-1)}$), 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 an integral point in $\mathcal{D}(P)$.

3 FDT on binary MILPs

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 $\min_{(x,y)\in S} cx$. In this section, we prove Theorem 2 by describing the Fractional Decomposition Tree (FDT) algorithm. We also remark that if $g = 1$, then the algorithm will give an exact decomposition of any feasible solution.

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. In Lemma 1 we saw how to 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).

$$
\begin{aligned}
\text{Branching}(\ell, x', y') \qquad \max \quad & \lambda_0 + \lambda_1 && && (10)\\
\text{s.t.} \quad & Ax^j + Gy^j \geq b\lambda_j && \text{for } j = 0,1 && (11)\\
& 0 \leq x^j \leq \lambda_j && \text{for } j = 0,1 && (12)\\
& x^0_\ell = 0, \ x^1_\ell = \lambda_1 && && (13)\\
& x^0 + x^1 \leq x' && && (14)\\
& \lambda_0, \lambda_1 \geq 0 && && (15)
\end{aligned}
$$

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 1. *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 $\sum_{i=1}^k \theta_i \tilde{x}^i \leq gx'$.

$$
x' \geq \sum_{i=1}^{k} \frac{\theta_i}{g} \tilde{x}^i = \sum_{i \in [k]: \tilde{x}^i_\ell = 0} \frac{\theta_i}{g} \tilde{x}^i + \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g} \tilde{x}^i \tag{16}
$$

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 (14) is satisfied by Inequality (16). Also, for $j = 0,1$ we have

$$
Ax^j + Gy^j = \sum_{i \in [k], \tilde{x}^i_\ell = j} \frac{\theta_i}{g} (A\tilde{x}^i + G\tilde{y}^i) \geq b \sum_{i \in [k], \tilde{x}^i_\ell = j} \frac{\theta_i}{g} = b\lambda_j. \tag{17}
$$

Hence, Constraints (11) holds. Constraint (13) 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 (14) 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. $\square$

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 $L_0, \ldots, L_t$ of pairs of $(x, y) \in \mathcal{D}(P)$ together with multipliers λ with the following properties for $i = 0, \ldots, t$: (a) If $(x, y) \in L_i$, then $x_j \in \{0,1\}$ for $j = 1, \ldots, i$, 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 $L_0, \ldots, L_i$. Let the solutions in L_i be (x^j, y^j) for $j = 1, \ldots, k$ and λ_j be their multipliers, respectively. For each $j = 1, \ldots, k$ 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^0_j, λ^1_j 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 $\lambda_j \lambda^0_j$, $\lambda_j \lambda^1_j$, respectively, for $j = 1, \ldots, k$. 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 $L' = \{[(x^1, y^1), \gamma_1], \ldots, [(x^{|L'|}, y^{|L'|}), \gamma_{|L'|}]\}$. The variables of Pruning(L') are scalar variables θ_j for each node j in L'.

$$\text{Pruning}(L') = \{\max \sum_{j=1}^{|L'|} \theta_j \ : \ \sum_{j=1}^{|L'|} \theta_j x^j_i \leq x^*_i \text{ for } i = 1, \ldots, t, \ \theta \geq 0\} \tag{18}$$

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 $\mathbb{R}^{|L'|}$, θ^* has to satisfy $|L'|$ linearly independent constraints at equality. However, there are only t constraints of type $\sum_{j=1}^{|L'|} \theta_j x_i^j \leq x_i^*$. Therefore, there are at most t coordinates of θ_j^* that are non-zero. Set L_{i+1} which consists of (x^j, y^j) for $j = 1, \ldots, |L'|$ 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 $\sum_{j=1}^{|L'|} \theta_j^* \geq \sum_{j=1}^{|L'|} \gamma_j \geq \frac{1}{g^{i+1}}$. □

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 concludes the description of the FDT algorithm and proves Theorem 2. See Algorithm 2 for a summary of the FDT algorithm.

Algorithm 2: Fractional Decomposition Tree Algorithm

Input: $P = \{(x, y) \in \mathbb{R}^{n+p} : Ax + Gy \geq b\}$ and $S = \{(x, y) \in P : x \in \{0,1\}^n\}$ such that $g = \max_{c \in \mathbb{R}_+^n} \frac{\min_{(x,y)\in S} cx}{\min_{(x,y)\in P} cx}$ is finite, $(x^*, y^*) \in P$

Output: $(z^i, w^i) \in S$ and $\lambda_i \geq 0$ for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \lambda_i = 1$, and $\sum_{i=1}^k \lambda_i z^i \leq g^k x^*$ for a $k \leq t$

$L^0 \leftarrow [(x^*, y^*), 1]$
for $i = 1$ ***to*** t **do**
  $L' \leftarrow \emptyset$
  for $[(x, y), \lambda] \in L^i$ **do**
    Apply Lemma 3 to obtain $[(\hat{x}^0, \hat{y}^0), \gamma_0]$ and $[(\hat{x}^1, \hat{y}^1), \gamma_1]$
    $L' \leftarrow L' \cup \{[(\hat{x}^0, \hat{y}^0), \lambda\gamma_0]\} \cup \{[(\hat{x}^1, \hat{y}^1), \lambda\gamma_1]\}$
  end
  Apply Pruning(L') to obtain L^{i+1}.
end
for $[(x, y), \lambda] \in L^t$ **do**
  Apply Algorithm 1 to (x, y) to obtain $(z, w) \in S$
  $F \leftarrow F \cup \{[(z, w), \lambda]\}$
end
return F

4 FDT for 2EC

In Section 3 our focus was on binary MILPs. 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 edges 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 each 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.

Theorem 4. *Let $G = (V, E)$ and x be an extreme point of $\text{2EC}(G)$. 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, $C \leq g_{\text{2EC}}^k$, where g_{2EC} is the integrality gap of the 2-edge-connected multigraph problem with respect to the formulation in (4).*

We do not know the exact value for g_{2EC}, but we know $\frac{6}{5} \leq g_{\text{2EC}} \leq \frac{3}{2}$ [4, 17]. Also, we need to remark that the polyhedral analysis of Christofides' algorithm provides a $\frac{3}{2}$-approximation for 2EC, i.e. we already have an algorithm with $C \leq \frac{3}{2}$ [17]. However, we will show in Section 5 that in practice the constant C for the FDT algorithm for 2EC is much better than $\frac{3}{2}$ for FEPs of order 10.

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. Arash: **put** $\hat{x}^1$

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

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

$$\max \quad \sum_{j=0,1,2} \lambda_j \tag{19}$$

$$\text{s.t.} \quad x^j(\delta(U)) \geq 2\lambda_j \qquad \text{for } \emptyset \subset U \subset V, \text{ and } j = 0,1,2 \tag{20}$$

$$0 \leq x^j \leq 2\lambda_j \qquad \text{for } j = 0,1,2 \tag{21}$$

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

$$x^j_f \geq \lambda_j \qquad \text{for } f \in E \text{ where } x_f \geq 1, \text{ and } j = 0,1,2 \tag{23}$$

$$x^0 + x^1 + x^2 \leq x \tag{24}$$

$$\lambda_0, \lambda_1, \lambda_2 \geq 0 \tag{25}$$

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 $\gamma_0 + \gamma_1 + \gamma_2 \geq \frac{1}{g_{2\text{EC}}}$, 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 (23) is also satisfied.

Claim 2. *We have* $\gamma_0 + \gamma_1 + \gamma_2 \geq \frac{1}{g_{2\text{EC}}}$.

Proof. Suppose for contradiction $\sum_{j=0,1,2} \gamma_j = \frac{1}{g_{2\text{EC}}} - \epsilon$ for some $\epsilon > 0$. Construct graph G' by removing edge f with $x_f \geq 1$ and replacing it with a path P_f of length $\lceil \frac{2}{\epsilon} \rceil$. Define $x'_h = x_h$ for each edge h such that $x_h < 1$. For each $h \in P_f$ let $x'_h = x_f$ for all f with $x_f \geq 1$. It is easy to check that $x' \in 2\text{EC}(G')$. By Theorem 1 there exists $\theta \in [0,1]^k$, with $\sum_{i=1}^k \theta_i = 1$ and 2-edge-connected multigraphs F'_i of G' for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i \chi^{F'_i} \leq g_{2\text{EC}} x'$.

Note that each F'_i contains at least one copy of every edge in any path P_f, except for at most one edge in the path. We will obtain 2-edge-connected multigraphs $F_1, \ldots, F_k$ of G using $F'_1, \ldots, F'_k$, respectively. To obtain F_i first remove all P_f paths from F'_i. Suppose there is an edge h in P_f such that $\chi^{F'_i}_h = 0$, this means that for any edge $p \in P_f$ such that $p \neq h$, $\chi^{F'_i}_p = 2$. In this case, let $\chi^{F_i}_f = 2$, i.e. add two copies of f to F_i. If there are at least two edges $h, h' \in P_f$ with $\chi^{F'_i}_h = \chi^{F'_i}_{h'} = 1$, let $\chi^{F_i}_f = 1$, i.e. add one copy of f to F_i. If for all edges $h \in P_f$, we have $\chi^{F'_i}_h = 2$, then let $\chi^{F_i}_f = 2$. For $f \in E$ with $x_f < 1$ we have

$$\sum_{i=1}^k \theta_i \chi^{F_i}_f = \sum_{i=1}^k \theta_i \chi^{F'_i}_f \leq g_{2\text{EC}} x'_f = g_{2\text{EC}} x_f. \tag{26}$$

In addition for $f \in E$ with $x_f \geq 1$ we have $\chi_f^{F_i} \leq \frac{\sum_{h \in P_f} \chi_h^{F_i'}}{\lceil \frac{2}{\epsilon} \rceil - 1}$ by construction.

$$\begin{aligned}
\sum_{i=1}^{k} \theta_i \chi_f^{F_i} &\leq \sum_{i=1}^{k} \theta_i \frac{\sum_{h \in P_f} \chi_h^{F_i'}}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&= \frac{\sum_{h \in P_f} \sum_{i=1}^{k} \theta_i \chi_h^{F_i'}}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&\leq \frac{\sum_{h \in P_f} g_{2\mathrm{EC}} x_h'}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&= \frac{\sum_{h \in P_f} g_{2\mathrm{EC}} x_f}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&= \frac{\lceil \frac{2}{\epsilon} \rceil}{\lceil \frac{2}{\epsilon} \rceil - 1} g_{2\mathrm{EC}} x_f
\end{aligned}$$

Therefore, we have

$$x_e \geq \sum_{i \in [k]: \chi_e^{F_i} = 1} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\mathrm{EC}} \lceil \frac{2}{\epsilon} \rceil} \chi_e^{F_i} + \sum_{i \in [k]: \chi_e^{F_i} = 2} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\mathrm{EC}} \lceil \frac{2}{\epsilon} \rceil} \chi_e^{F_i}. \tag{27}$$

Let x^j be such that $x_e^j = \sum_{i \in [k]: \chi_e^{F_i} = j} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\mathrm{EC}} \lceil \frac{2}{\epsilon} \rceil} \chi^{F_i}$ and $\theta_j = \sum_{i \in [k]: \chi_e^{F_i} = j} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\mathrm{EC}} \lceil \frac{2}{\epsilon} \rceil}$ for $j = 0, 1, 2$. It is easy to check that x^j , θ_j for $j = 0, 1, 2$ is a feasible solution to the LP above. Notice that $\sum_{j=0,1,2} \theta_j = \frac{\lceil \frac{2}{\epsilon} \rceil - 1}{g_{2\mathrm{EC}} \lceil \frac{2}{\epsilon} \rceil}$. By assumption, we have $\frac{\lceil \frac{2}{\epsilon} \rceil - 1}{g_{2\mathrm{EC}} \lceil \frac{2}{\epsilon} \rceil} \leq \frac{1}{g_{2\mathrm{EC}}} - \epsilon$. However, this means $2 \leq 1$ which is a contradiction. □

This concludes the proof. □

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) $\sum_{(x,\lambda) \in L_i} \lambda \geq \frac{1}{g_{2\mathrm{EC}}^i}$, (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 (v) in Lemma 5 to prove that (a) also holds.

Proof. We proceed by induction on i. Define $L_0 = \{(x^*, 1)\}$. It is easy to check all the properties are satisfied. Now, suppose by induction we have L_{i-1} for some $i = 1, \ldots, t$ that satisfies all the properties. For each solution x^ℓ in L_{i-1} apply Lemma 5 on x^ℓ and e_i to obtain $x^{\ell j}$ and $\lambda_{\ell j}$ for $j = 0, 1, 2$. Let L' be the collection that contains $(x^{\ell j}, \lambda_\ell \cdot \lambda_{\ell j})$ for $j = 0, 1, 2$, when applied to all (x^ℓ, λ_ℓ) in L_{i-1}. Similar to the proof in Lemma 4 one can check that L_i satisfies properties (b), (c). We now verify property (a). Consider a solution x^ℓ in L_{i-1}. For $e \in \{e_1, \ldots, e_{i-1}\}$ if $x^\ell_e = 0$, then by property (iv) in Lemma 5 we have $x^{\ell j}_e = 0$ for $j = 0, 1, 2$. Otherwise by induction we have $x^\ell_e \geq 1$ in which case property (v) in Lemma 5 ensures that $x^{\ell j}_e \geq 1$ for $j = 0, 1, 2$. Also, $x^{\ell j}_{e_i} = j$, so $x^{\ell j}_{e_i} = 0$ or $x^{\ell j}_{e_i} \geq 1$ for $j = 0, 1, 2$.

Finally, if $|L'| \leq t$ we let $L_i = L'$, otherwise apply Pruning(L') to obtain L_i with $|L_i| \leq t$. □

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 $t \leq 2|V| - 1$ [7]. By Lemma 6 we have

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

5 Computational experiments with FDT

We ran FDT on two covering problems: the tree augmentation problem (TAP) and 2EC. In TAP we are given a tree $T = (V, E)$, and a set of non-edges L between vertices in V and costs $c \in \mathbb{R}^L_{\geq 0}$. A feasible augmentation is $L' \subseteq L$ such that $T + L'$ is 2-edge-connected. In TAP we wish to find the minimum-cost feasible augmentation.

For $\ell \in L$, let P_ℓ be the set of edges in the unique path between the endpoints of ℓ in T. Let $\mathrm{TAP}(T, L) = \{x \in [0,1]^L : \sum_{\ell : e \in P_\ell} x_\ell \geq 1, \text{ for } e \in E\}$. Then $\mathrm{TAP}(T, L) \cap \mathbb{Z}^L$ is set of feasible augmentations for the instance defined by T and L. We know $\frac{3}{2} \leq g_{\mathrm{TAP}} \leq 2$ [10, 6]. We ran binary FDT on a set of 264 fractional extreme points of randomly produces instances of TAP. Table 1 shows FDT found solutions better than the integrality-gap lower bound for most instances.

	$C \in [1.1, 1.2]$	$C \in (1.2, 1.3]$	$C \in (1.3, 1.4]$	$C \in (1.4, 1.5]$
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.

We also implemented the polyhedral version of Christofides' algorithm [17]. In particular, we implemented the T-join augmentation in Christofides' algorithm, in a way that minimizes the average usage of every edge across the the convex combination of spanning trees. This allows the "real" approximation ratio of Christofides' algorithm to be below $\frac{3}{2}$. Figure 1 shows FDT's solutions on all FEPs of order 10 are always better than those from Christofides' algorithm.

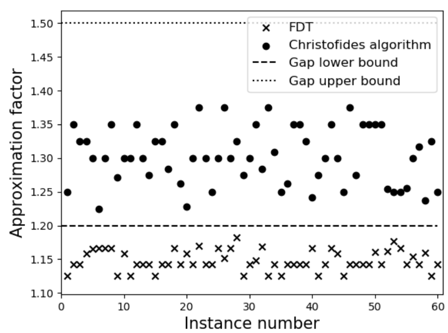

Figure 1: Christofides' algorithm vs FDT on all fundamental extreme points of order 10.

We ran FDT for 2EC 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 $\frac{6}{5}$.

6 Strengthening Theorem 1 for bounded covering problems

Theorem 1 characterizes the integrality gap as the smallest number such that for any point $x \in \mathcal{D}(P)$, gx dominates a convex combination of points in S, i.e. there exists convex combination $\sum_{i=1}^{k} \lambda_i x^i \leq gx$. In a covering problem, we assume that matrix A in the description of P is a non-negative matrix, hence if $y \geq x$ and $x \in P$ we have $y \in P$. We also assume the right hand side vector in (2) is of the form $b\mathbf{1}$, i.e. it is a uniform vector. Finally we assume we have bound constraints $x \leq b\mathbf{1}$. This class of problems include a broad class of problems. Notice that for a covering problem $\mathcal{D}(P) = P$. We show that for these problems, we can make Theorem 1 stronger in the following sense.

Theorem 5. *Let $x \in P$, there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^{k} \theta_i = 1$ and $\tilde{x}^i \in S$ for $i = 1, \dots, k$ such that*

- *for $\ell \in \{1, \dots, n\}$, if $x_\ell = 0$, then $\tilde{x}^i_\ell = 0$ for $i = 1, \dots, k$, i.e. $\tilde{x}^i$ is in the support of x,*
- *for $j \in \{1, \dots, m\}$ with $A_j x = b$, we have $A_j(\sum_{i=1}^{k} \theta_i \tilde{x}^i) \leq C \cdot A_j x$,*

if and only if $C \geq g$.

In other words, the integrality gap is the smallest number such that for any $x \in P$, there exists a convex combination $y = \sum_{i=1}^{k} \lambda_i x^i$, such that $A_j y \leq g(A_j x)$ for all j such that $A_j x = b$.

Notice that the proof of this theorem, same as Theorem 1, does not imply a polynomial time algorithm.

Observe that if $C \geq g$, by Theorem 1, there is a convex combination of points in S that is dominated by Cx, i.e. there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^{k} \theta_i = 1$, and $x^i \in S$, such that $\sum_{i=1}^{k} \theta_i x^i \leq Cx$. The same convex combination also trivially satisfies both requirements in Theorem 5. The more interesting direction is the converse. We begin with the following theorem. Suppose $v^1, \dots, v^r$ are the extreme points of P. Assume for each $i = 1, \dots, r$, there exists a convex combination of points in S, namely $\sum_{\ell=1}^{k_i} \lambda^i_\ell w^i_\ell$ such that $A_j \sum_{\ell=1}^{k_i} \lambda^i_\ell w^i_\ell \leq C A_j v^i$ for j such that $A_j v^i = b$. Let $z^i = \sum_{\ell=1}^{k_i} \lambda^i_\ell w^i_\ell$.

Lemma 8. *For any $i \in \{1, \dots, r\}$, there exists $\epsilon_i > 0$ and $x^i \in P$ where $x^i \leq \frac{Cv^i - \epsilon_i z^i}{C(1-\epsilon_i)}$.*

Bob: $u^1(\epsilon)$ **maybe change to** $x^i(\epsilon)$

Proof. Let $x^i(\epsilon) = \frac{Cv^i - \epsilon z^i}{C(1-\epsilon)}$. For $j \in \{1, \dots, m\}$, let $t_j = A_j z^i - Cb$ and $s_j = A_j v^i - b$. Assume $\epsilon < 1$. For $j \in \{1, \dots, m\}$ such that $A_j z^i \leq Cb$, i.e. $t_j \leq 0$ (this includes j such that $A_j v^i = b$, i.e. $s_j = 0$) we have

$$A_j x^i(\epsilon) = \frac{CA_j v^i - \epsilon A_j z^i}{C(1-\epsilon)} \geq \frac{Cb - \epsilon Cb}{C(1-\epsilon)} = b$$

Assume $0 < \epsilon \leq C \cdot \min_{j:t_j>0} \frac{s_j}{t_j}$. Notice that $\min_{j:t_j>0} \frac{s_j}{t_j} > 0$ since for j such that $t_j > 0$, we have $A_j z^i > Cb$ which implies that $A_j v^i > b$, so $s_j > 0$. For j such that $t_j > 0$ we have

$$\begin{aligned}
A_j x^i(\epsilon) &= \frac{CA_j v^i - \epsilon A_j z^i}{C(1-\epsilon)} \\
&= \frac{C(b+s_j) - \epsilon(Cb+t_j)}{C(1-\epsilon)} \\
&= b + \frac{Cs_j - \epsilon t_j}{C(1-\epsilon)} \\
&\geq b \qquad\qquad (\epsilon \leq C \cdot \frac{s_j}{t_j}, \text{ and } \epsilon < 1, \text{ so } \frac{Cs_j - \epsilon t_j}{C(1-\epsilon)} \geq 0)
\end{aligned}$$

Therefore if $\epsilon \leq 1$ and $\epsilon \leq C \cdot \min_{j:t_j>0} \frac{s_j}{t_j}$ we have $Ax^i(\epsilon) \geq \mathbf{1}b$. Next, we show that we can choose ϵ so that $x^i(\epsilon) \geq 0$.

For $\ell \in \{1, \ldots, n\}$, if $z^i_\ell = 0$, then we have $x^i_\ell(\epsilon) = \frac{Cv^i_\ell}{C(1-\epsilon)} \geq 0$, since $v^i_\ell \geq 0$ and $\epsilon < 1$. Otherwise we have $z^i_\ell > 0$. Assume $\epsilon \leq C \cdot \min_{\ell:z^i_\ell>0} \frac{v^i_\ell}{z^i_\ell}$. Notice that for ℓ such that $z^i_\ell > 0$, we have $v^i_\ell > 0$ since z^i lies in the support of v^i by assumption. This means that $\min_{\ell:z^i_\ell>0} \frac{v^i_\ell}{z^i_\ell} > 0$. Now, for ℓ such that $z^i_\ell > 0$, we have $x^i_\ell(\epsilon) = \frac{Cv^i_\ell - \epsilon z^i_\ell}{C(1-\epsilon)} \geq 0$ by choice of ϵ.

Define $\epsilon_i = \frac{1}{2} \cdot \min\{1, C \cdot \min_{j:t_j>0} \frac{s_j}{t_j}, C \cdot \min_{\ell:z^i_\ell>0} \frac{v^i_\ell}{z^i_\ell}\}$. By the arguments above $Ax^i(\epsilon_i) \geq \mathbf{1}b$ and $x^i(\epsilon_i) \geq 0$. Define x^i as follows: For each $j \in \{1, \ldots, m\}$, if $x^i_j(\epsilon_i) \leq b$, then let $x^i_j = x^i_j(\epsilon_i)$, otherwise let $x^i_j = b$. We need to show that $x^i \in P$. Note that clearly $x^i \geq 0$, hence we only need to show that $Ax^i \geq \mathbf{1}b$. Suppose for contradiction there is $j \in \{1, \ldots, m\}$ such that $A_j x^i < b$. Since $x^i(\epsilon) \in P$, there must be $\ell^* \in \{1, \ldots, n\}$ such that $x^i_{\ell^*}(\epsilon) > b$, and $A_{j,\ell^*} > 0$. But this means that $A_{j,\ell^*} \geq 1$ and $x^i_{\ell^*} = b$. Therefore $A_j x^i \geq A_{j,\ell^*} x^i_{\ell^*} = A_{j,\ell^*} b \geq b$. $\square$

Lemma 9. *There exists a convex combination of point in S that is dominated by Cv^i for $i = 1, \ldots, r$.*

Proof. We prove something slightly stronger. We show that for $i \in \{1, \ldots, r\}$, and for $j = 1, \ldots, i$, the vector Cv^j dominates a convex combination of $z^1, \ldots, z^i$ and $Cv^{i+1}, \ldots, Cv^r$. Note that this is enough to prove our statement since for $i = r$, the statement implies that Cv^j dominates a convex combination of $z^1, \ldots, z^r$ for $j = 1, \ldots, r$.

We proceed by induction on i. If $i = 1$, we just need to show that Cv^1 dominates by a convex combination of z^1 and $Cv^2, \ldots, Cv^r$. By Lemma 8, there exists $x^1 \leq \frac{Cv^1 - \epsilon_1 z^1}{C(1-\epsilon_1)}$. We have $x^1 \in P$, so we can write x^1 as convex combination of $v^1, \ldots, v^r$. We have

$$Cv^1 \geq \epsilon_1 z^1 + C(1-\epsilon_1)x^1 \geq \epsilon_1 z^1 + C(1-\epsilon_1) \sum_{\ell=1}^{r} \lambda^1_\ell v^\ell. \tag{28}$$

Hence,

$$Cv^1 \geq \frac{\epsilon_1}{1-(1-\epsilon_1)\lambda_1^1} \cdot z^1 + (1-\epsilon_i) \sum_{\ell=2}^{r} \frac{\lambda_\ell^1}{1-(1-\epsilon_1)\lambda_1^1} \cdot Cv^\ell. \tag{29}$$

A curious reader might notice that Inequality 28 looks very strange: Based on Lemma 8 we can consider the limit of this inequality when ϵ_1 goes to zero. However, assuming $\lim_{\epsilon\to 0} \frac{\epsilon_1}{1-(1-\epsilon_1)\lambda_1^1} = 0$, and $\lim_{\epsilon\to 0} \frac{\lambda_\ell^1}{1-(1-\epsilon_1)\lambda_1^1}$ exists for $\ell = 2, \ldots, r$, we would have $v^1 \geq \sum_{\ell=2}^{r} \frac{\lambda_\ell^1}{1-\lambda_1^1} v^\ell$. This is a contradiction to the fact that v^1 is an extreme point. We will later illustrate with an example where $\lim_{\epsilon\to 0} \frac{\epsilon_1}{1-(1-\epsilon_1)\lambda_1^1}$ exists and is not zero (See Example 1)

Observe that $\frac{\epsilon_1}{1-(1-\epsilon_1)\lambda_1^1}$ and $(1-\epsilon_1)\frac{\lambda_\ell^1}{1-(1-\epsilon_1)\lambda_1^1}$ for $\ell = 2, \ldots, r$ form convex multipliers, so the base case holds.

Now consider $i \in \{1, \ldots, r-1\}$. By induction, for $j = 1, \ldots, i$ we have

$$Cv^j \geq \sum_{\ell=1}^{i} \theta_\ell^j z^\ell + \sum_{\ell=i+1}^{r} \theta_\ell^j Cv^\ell. \tag{30}$$

Suppose we are able to show

$$Cv^{i+1} \geq \sum_{\ell=1}^{i+1} \mu_\ell z^\ell + \sum_{\ell=i+2}^{r} \mu_\ell Cv^\ell, \tag{31}$$

where $\mu \geq 0$, and $\sum_{\ell=1}^{r} \mu_\ell = 1$. Then for $j = 1, \ldots, i$ we have

$$\begin{aligned}
Cv^j &\geq \sum_{\ell=1}^{i} \theta_\ell^{j,i} z^\ell + \theta_{i+1}^{j,(i+1)} Cv^{i+1} + \sum_{\ell=i+2}^{r} \theta_\ell^j Cv^\ell && \text{(By 30)} \\
&\geq \sum_{\ell=1}^{i} \theta_\ell^j z^\ell + \sum_{\ell=1}^{i+1} \theta_{i+1}^j \mu_\ell z^\ell + \sum_{\ell=i+2}^{r} \theta_{i+1}^j \mu_\ell Cv^\ell + \sum_{\ell=i+2}^{r} \theta_\ell^j Cv^\ell && \text{(By 31)} \\
&= \sum_{\ell=1}^{i} (\theta_\ell^j + \mu_\ell \theta_{i+1}^j) z^\ell + \mu_{i+1} \theta_{i+1}^j z^{i+1} + \sum_{\ell=i+2}^{r} (\theta_\ell^j + \mu_\ell \theta_{i+1}^j) Cv^\ell.
\end{aligned}$$

It is easy to see that the multipliers above are convex. Thus, we only need to show the convex combination in (31) exists. This will be similar to what we did in the base case. By Lemma 8, there exists ϵ_{i+1} and x^{i+1} such that

$$\begin{aligned}
Cv^{i+1} &\geq \epsilon_{i+1} z^{i+1} + C(1-\epsilon_{i+1}) x^{i+1} \\
&\geq \epsilon_{i+1} z^{i+1} + (1-\epsilon_{i+1}) \sum_{\ell=1}^{k} \lambda_\ell Cv^\ell
\end{aligned}$$

Applying induction we can replace Cv^j by $\sum_{\ell=1}^{i} \theta_\ell^j z^\ell + \sum_{\ell=i+1}^{r} \theta_\ell^j C v^\ell$ for $j = 1, \ldots, i$. This means

$$(1-(1-\epsilon_{i+1})(\lambda_{i+1}+\sum_{j=1}^{i}\lambda_j\theta_{i+1}^j))Cv^{i+1} \geq \epsilon_{i+1}z^{i+1} + (1-\epsilon_{i+1})\sum_{j=1}^{i}\lambda_j(\sum_{\ell=1}^{i}\theta_\ell^j z^\ell + \sum_{\ell=i+2}^{r}\theta_\ell^j Cv^\ell)$$
$$+ (1-\epsilon_{i+1})(\sum_{j=1}^{i}\lambda_j\lambda_{i+1}^j + \lambda_{i+1}^i)C + (1-\epsilon_{i+1})\sum_{j=i+2}^{r}\lambda_j Cv^j$$

It is easy to show that multipliers on the RHS above are convex after multiplying be the positive constant on the LHS. □

Example 1. Consider $2\mathrm{EC}(G) = \{x \in [0,2]^E : \; x(\delta(U)) \geq 2 \text{ for } \emptyset \subset U \subset V\}$, where $G = (V, E)$ is the graph in Figure 2.

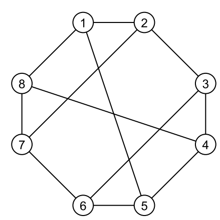

Figure 2: Graph $G = (V, E)$. Let H be the Hamiltonian cycle of G that contains edges $(1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (1,8)$. Let $M = E \setminus H$: $M = \{(1,5), (2,7), (3,6), (4,8)\}$.

Let v^1 be the following extreme point of $2\mathrm{EC}(G)$:

$$v_e^1 = \begin{cases} 1 & \text{if } e \in M; \\ 1/2 & \text{if } e \in H. \end{cases}$$

Define z^1 to be the following vector:

$$z_e^1 = \begin{cases} 7/5 & \text{if } e = (1,5); \\ 1 & \text{if } e \in M \setminus \{(1,5)\}; \\ 2/5 & \text{if } e \in \{(1,8), ((4,5))\}; \\ 3/5 & \text{if } e \in H \setminus \{(1,8), ((4,5))\}. \end{cases}$$

It is easy to check that z^1 satisfies the requirement of Theorem 6 when setting $C = \frac{6}{5}$. In particular, we have $z^1(\delta(U)) \leq \frac{6}{5}v^1(\delta(U))$ for all $\emptyset \subset U \subset V$. Also, is it easy to decompose z^1 into a convex combination of integer point in of $2\mathrm{EC}(G)$.

Now, we want to set the ϵ_i in the statement of Lemma 8 to be 2^{-k}. By Lemma 8

$$x_e^1 = \begin{cases} \frac{2^k - 7/6}{2^k - 1} & \text{if } e = (1,5); \\ \frac{2^k - 5/6}{2^k - 1} & \text{if } e \in M \setminus \{(1,5)\}; \\ \frac{2^k - 2/3}{2 \cdot (2^k - 1)} & \text{if } e \in \{(1,8), ((4,5))\}; \\ \frac{1}{2} & \text{if } e \in H \setminus \{(1,8), ((4,5))\}. \end{cases}$$

is in 2EC(G) for sufficiently large k. The vectors defined below $v^2, \ldots, v^7$ are also integer extreme points of 2EC(G).

	(1,5)	(2,7)	(3,6)	(4,8)	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)	(6,7)	(7,8)	(1,8)
$v^2 =$	(0	1	2	1	0	1	1	2	0	0	1	2)
$v^3 =$	(0	2	2	1	0	0	0	1	1	1	1	2)
$v^4 =$	(1	2	2	2	1	1	1	1	0	0	0	0)
$v^5 =$	(0	2	1	1	1	1	0	1	1	0	0	1)
$v^6 =$	(1	1	1	2	0	1	0	0	1	0	1	1)
$v^7 =$	(1	1	1	2	1	0	1	1	0	1	0	0)

Moreover, we have

$$x^1 = (1 - 8\lambda)v^1 + \lambda v^2 + 2\lambda v^3 + \lambda v^4 + \lambda v^5 + \lambda v^6 + 2\lambda v^7, \quad \lambda = \frac{1}{24 \cdot (2^k - 1)}$$

Now, this allows us to rewrite Inequality (29) for this example.

$$\frac{6}{5}v^1 \geq \frac{3}{4}z^1 + \frac{1}{32}(\frac{6}{5} \cdot v^2) + \frac{1}{16}(\frac{6}{5} \cdot v^3) + \frac{1}{32}(\frac{6}{5} \cdot v^4) + \frac{1}{32}(\frac{6}{5} \cdot v^5) + \frac{1}{32}(\frac{6}{5} \cdot v^6) + \frac{1}{16}(\frac{6}{5} \cdot v^7).$$

References

[1] 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.

[2] Geneviève Benoit and Sylvia Boyd. Finding the exact integrality gap for small traveling salesman problems. *Mathematics of Operations Research*, 33(4):921–931, 2008.

[3] 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.

[4] 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.

[5] 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.

[6] Joseph Cheriyan, Howard Karloff, Rohit Khandekar, and Jochen Könemann. On the integrality ratio for tree augmentation. *Operations Research Letters*, 36(4):399–401, 2008.

[7] 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.

[8] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, Sep 2005.

[9] Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1(2):201–222, Oct 2009.

[10] Greg N. Frederickson and Joseph Ja'Ja'. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981.

[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] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[15] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

[16] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[17] Laurence A. Wolsey. *Heuristic analysis, linear programming and branch and bound*, pages 121–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 1980.