

FDT

BOB, CINDY, ARASH

October 12, 2018

Abstract

TO ADD.

1 Introduction

Mixed-integer linear programming (MILP), the optimization of a linear objective function subject to linear and integrality constraints, is a classic NP-hard problem [GJ79]. In fact it is NP-hard in general to determine if there is a feasible solution to an MILP problem [GJ79]. However, intelligent branch-and-bound strategies allow commercial and open-source MIP solvers to give exact solutions (or near-optimal with provable bound) to many specific instances of NP-hard combinatorial optimization problems. Modeling combinatorial optimization problems with MILPs also started the very popular area of LP-based approximation algorithms.

Consider the set of point described by a set

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

and the linear relaxation of set S

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

To make the notation simpler we use $I = (A, G, b)$, i.e. an instance of the formulation. Then we use $S(I)$ and $P(I)$ to denote $S(A, G, b)$ and $P(A, G, b)$, respectively. Notice that $\min \{cx : (x, y) \in S(I)\}$ is in fact an MILP problem and $\min \{cx : (x, y) \in P(I)\}$ is lower bound provided from the LP relaxation. Let $z_{IP}(I, c)$ and $z_{LP}(I, c)$ be the optimal solution to these problems, respectively.

In many papers ([1]) researchers have developed polynomial time algorithms that find solutions for special classes of MILPs whose cost are provably smaller than $C \cdot z_{LP}(I, c)$ for some C which can be a constant number or dependent on some parameter of the MILP, e.g. $\log(n)$. However, for many combinatorial optimization problems there is a limit to

such techniques. Let $g_I = \max_{c \geq 0} \frac{z_{IP}(I, c)}{z_{LP}(I, c)}$. Parameter g_I depends on set S and the linear constraints in (1) and is known as the *integrality gap* of the formulation for instance I . It is easy to see, that we cannot hope to find solutions for the MILP with objective values better than $g \cdot z_{IP}(I, c)$. More generally we can define the integrality gap for a class of instances \mathcal{I} . In this case, the integrality gap for problem \mathcal{I} is

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

For example, in the problem of finding a minimum weight 2-edge-connected multigraph, the gap of the the natural formulation, constraining every cut to be crossed at least twice, is at least $\frac{6}{5}$. Therefore, we cannot hope to obtain an LP-based $(\frac{6}{5} - \epsilon)$ -approximation algorithm for this problem using this LP relaxation. If we have $g_I = 1$, $P(I) = \text{conv}(S(I))$ and $P(I)$ is an *integral* polyhedron. There are many interesting examples of such formulations such as spanning tree polytope and the perfect matching polytope [Sch03]. For such problems we know how to write a vector $x \in P(I)$ as convex combination of points in $S(I)$ in polynomial time [?].

Fact 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 $i = 1, \dots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq x$. Moreover, we can find such a convex combination in polynomial time.*

Now assume $1 < g < \infty$. A analogous result to Fact 1 is the following theorem due to Carr and Vempala [CV04].

Theorem 1 (Carr, Vempala [CV04]). *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 $i = 1, \dots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq Cx$ if and only if $g_I \leq C$.*

In contrast to Fact 1 which implies exact algorithms for problems with a gap of 1, Theorem 1 does not imply an approximation algorithm, since it does not suggest how to find such a convex combination in polynomial time. This points to an interesting open question.

Question 1. *Assume reasonable complexity assumptions (such as UGC or $P \neq NP$). Given instance I with $1 < g_I < \infty$ and $(x, y) \in P(I)$, can we find $\theta \in [0, 1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $(\tilde{x}^i, \tilde{y}^i) \in \mathcal{D}(S(I))$ for $i = 1, \dots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq g_I x$ in polynomial time?*

This seems to be a very hard question. A more specific question is of more interest.

Question 2. *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 $(\tilde{x}^i, \tilde{y}^i) \in S$ for $i = 1, \dots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq g_{\mathcal{I}} x$ in polynomial time?*

Although Question 2 is wide open, for some problems \mathcal{I} there are polynomial time algorithms. For example, for generalized Steiner forest problem [?] gave an LP-based 2-approximation algorithm. The gap for this problem is also lower bounded by 2. Same holds for the set covering problem []. In fact, for set cover the approximation algorithm achieving the same factor as the integrality gap lower bound, is a *randomized rounding* algorithm. Rhagavan and Thompson [?] showed that this technique achieves provably good approximation for many combinatorial optimization problems.

One key step, in answering Question 1 and 2 is being able to find a solution in $S(I)$. In fact, 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 [?, ?]. These heuristics are often very effective and fast in practice, however, they can sometimes fail in finding a feasible solution. In addition these algorithms do not provide any bounds on the quality of the solution they find.

1.1 Our results

In this paper we study Theorem 1 and the natural questions that follow from it (Questions 1 and 2).

We give a general approximation framework for solving $\{0, 1\}$ -MILPs. Consider the set of point described by a set

$$S(I) = \{(x, y) \in \mathbb{R}^{n \times p} : Ax + Gy \geq b, x \in \{0, 1\}\}, \quad (4)$$

and a linear relaxation of set S

$$P(I) = \{(x, y) \in \mathbb{R}^{n \times p} : Ax + Gy \geq b, 0 \leq x \leq 1\}. \quad (5)$$

For a point $(x, y) \in P(I)$, let $\text{sup}(x, y) = \{i \in \{1, \dots, n\} : x_i \neq 0\}$. We prove the following algorithm.

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

Notice, that the FDT algorithm needs to find feasible solutions to any MILP with finite gap. This can be of independent interest. In the following theorem $\mathcal{D}(P(I)) = \{(x', y') : (x', y') \geq (x, y) \text{ for some } (x, y) \in P(I)\}$

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

In attempt to extend our results into $\{0, 1, 2\}$ -MILPs we give an FDT algorithm for the 2-edge-connected multigraph problem. In this problem we want to find the minimum weight 2-edge-connected multigraph in a graph $G = (V, E)$ with respect to weights $c \in \mathbb{R}_{\geq 0}^E$. For this problem the natural linear programming relaxation is

$$2EC(G) = \{x \in \mathbb{R}_{\geq 0}^E : x(\delta(S)) \geq 2 \text{ for } \emptyset \subset S \subset V\}. \quad (6)$$

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, \dots, F_k such that $k \leq 2|V|$, $\sum_{i=1}^k \lambda_i \chi^{F_i} \leq Cx^*$, and $\sum_{i=1}^k \lambda_i = 1$. Moreover, $C \leq g_{2EC}^{2|V|}$, where g_{2EC} is the integrality gap of the 2-edge-connected multigraph problem with respect to formulation in (6)*

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 providing approximation ratios that are far better than the theoretical bound in the theorem statements. We examine FDT for binary MILPs for problems such as the tree augmentation problem (TAP) and we apply FDT for 2EC on some interesting and “hard to decompose” points in the linear relaxation.

Concerning Theorem 1 we give a stronger characterization of integrality gap for covering problems. For this purpose assume matrix $A \in \mathbb{R}_{\geq 0}^n$ and $p = 0$, e.g. we have pure integer program which is also a covering problem.

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

- *for $j = 1, \dots, n$, if $x_j = 0$, then $\tilde{x}_j^i = 0$ for $i = 1, \dots, k$, i.e. x^i is in the support of x ,*
- *we have $Cb = C \cdot A_\ell x \geq A_\ell(\sum_{i=1}^k \theta_i \tilde{x}^i)$, for ℓ such that $A_\ell x = b$,*

if and only if $C \geq g_I$.

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

1.2 Notation and Preliminaries

2 FDT on binary MIPs

In this section we present the fractional decomposition tree (FDT) algorithm and prove its correctness.

Consider a formulation instance $I = (A, G, b)$, set

$$S(I) = \{(x, y) \in \mathbb{R}^{n \times p} : Ax + Gy \geq b, x \in \{0, 1\}\}, \quad (7)$$

and its linear relaxation

$$P(I) = \{(x, y) \in \mathbb{R}^{n \times p} : Ax + Gy \geq b, 0 \leq x \leq 1\}. \quad (8)$$

Also assume we are given a point $(x^*, y^*) \in P(I)$. 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 the following theorem.

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

One key ingredient in proving Theorem 2 is Theorem 3 that we will also prove in this section.

Although the theoretical worst-case upper bound on C in the statement of Theorem 2 can be very large, we will show that in practice C can be really small and hence FDT can provide good LP-based approximation algorithms in many cases. We also remark that if $g_I = 1$, then the algorithm will give an exact decomposition of any feasible solution. In the remainder of this section we explain the FDT algorithm and prove Theorem 2. For simplicity in the notations we denote $P(I)$, $S(I)$, and g_I with P , S , and g , respectively. Let $t = |\sup(x^*, y^*)|$. We assume without loss of generality that $\sup(x^*, y^*) = \{1, \dots, t\}$.

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

Lemma 1. *Given $(x', y') \in \mathcal{D}(P)$ and $\ell \in 1, \dots, n$, we can find in polynomial time vector (\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) $\hat{x}_\ell^0 = 0$ and $\hat{x}_\ell^1 = 1$,
- (iv) $\gamma_0 \hat{x}^0 + \gamma_1 \hat{x}^1 \leq x'$.

Proof of Lemma 1. Consider the following linear programing which we denote by $\text{Branching}(\ell, x', y')$.

The variables of $\text{Branching}(\ell, x', y')$ are γ_0, γ_1 and (x^0, y^0) and (x^1, y^1) .

$$\text{Branching}(\ell, x', y') \quad \max \quad \lambda_0 + \lambda_1 \quad (9)$$

$$\text{s.t.} \quad Ax^j + Gy^j \geq b\lambda_j \quad \text{for } j = 0, 2 \quad (10)$$

$$0 \leq x^j \leq \lambda_j \quad \text{for } j = 0, 1 \quad (11)$$

$$x_\ell^0 = 0, \quad x_\ell^1 = \lambda_1 \quad (12)$$

$$x^0 + x^1 \leq x' \quad (13)$$

$$\lambda_0, \lambda_1 \geq 0 \quad (14)$$

Let $(x^0, y^0), (x^1, y^1)$, and γ_0, γ_1 be an optimal solution to the LP above. Let $(\hat{x}^0, \hat{y}^0) = (\frac{x^0}{\gamma_0}, \frac{y^0}{\gamma_0})$, $(\hat{x}^1, \hat{y}^1) = (\frac{x^1}{\gamma_1}, \frac{y^1}{\gamma_1})$. Observe that (ii), (iii), (iv) are satisfied with this choice. In order 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 $i = 1, \dots, k$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq gx'$.

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

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

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

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

This concludes the proof. \square

We now show if x' in the statement of Lemma 1 is partially integral, we can find solutions with more integral components.

Lemma 2. Given $(x', y') \in \mathcal{D}(P)$, such that $x'_1, \dots, x'_{\ell-1} \in \{0, 1\}$ for some $\ell \geq 1$, we can find in polynomial time vector $(\hat{x}^0, \hat{y}^0), (\hat{x}^1, \hat{y}^1)$ and scalars $\gamma_0, \gamma_1 \in [0, 1]$ such that

$$(i) \quad \gamma_0 + \gamma_1 \geq \frac{1}{g},$$

- (ii) (\hat{x}^0, \hat{y}^0) and (\hat{x}^1, \hat{y}^1) are in $\mathcal{D}(P)$,
- (iii) $\hat{x}_\ell^0 = 0$ and $\hat{x}_\ell^1 = 1$,
- (iv) $\gamma_0 \hat{x}^0 + \gamma_1 \hat{x}^1 \leq x'$,
- (v) $\hat{x}_j^i \in \{0, 1\}$ for $i = 0, 1$ and $j = 1, \dots, \ell - 1$.

Proof. By Lemma 1 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 $j = 1, \dots, \ell - 1$, let $\hat{x}_j^i = \lceil \bar{x}_j^i \rceil$, for $j = \ell, \dots, n$ let $\hat{x}_j^i = \bar{x}_j^i$. 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 $\gamma_0 \hat{x}_j^0 + \gamma_1 \hat{x}_j^1 \leq g x'_j$. If $j = \ell, \dots, n$, 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 $\bar{x}_j^0 = \bar{x}_j^1 = 0$. Therefore, $\hat{x}_j^i = 0$ for $i = 0, 1$, so (iv) holds. Otherwise if $x'_j = 1$, then we have

$$\gamma_0 \hat{x}_j^0 + \gamma_1 \hat{x}_j^1 \leq \gamma_0 + \gamma_1 \leq 1 \leq x'_j.$$

Therefore (v) holds. □

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. Leaves of the FDT tree contain solutions with integer values for all the x variables that dominate a point in S . We will later see how we can turn these into points in S .

Let us define the FDT algorithm more formally. The 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, or L_t , are points in $\mathcal{D}(P)$ and are integral for all integer variables.

Lemma 3. *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 $j \in 1, \dots, k$ by Lemma 2 (setting $(x', y') = (x^j, y^j)$ and $\ell = i$) 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 2. Define L' to be the set of node with solutions $(x^{j0}, y^{j0}), (x^{j1}, y^{j1})$ and multipliers $\lambda_j \lambda_j^0, \lambda_j \lambda_j^1$, respectively, for $j = 1, \dots, \ell$.

Notice that for each $j \in \{1, \dots, k\}$ by property (v) in Lemma 2 we have $x_\ell^{j0}, x_\ell^{j1} \in \{0, 1\}$ for $\ell = 0, \dots, i + 1$. We also have

$$\begin{aligned}
\sum_{[(x,y),\lambda] \in L'} \lambda &= \sum_{j=1}^k \lambda_j \lambda_j^0 + \lambda_j \lambda_j^1 \\
&\geq \sum_{j=1}^k \frac{\lambda_j}{g} && \text{(By property (i) in Lemma 2)} \\
&\geq \frac{1}{g} \sum_{[(x,y),\lambda] \in L_i} \lambda \\
&\geq \frac{1}{g^{i+1}}. && \text{(By induction hypothesis)}
\end{aligned}$$

Also

$$\begin{aligned}
\sum_{[(x,y),\lambda] \in L'} \lambda x &= \sum_{j=1}^k \lambda_j (\lambda_j^0 x^{j0} + \lambda_j^1 x^{j1}) \\
&\leq \sum_{j=1}^k \lambda_j x^j && \text{(By property (iv) in Lemma 2)} \\
&\leq \sum_{[(x,y),\lambda] \in L_i} \lambda x \\
&\leq x^*. && \text{(By induction hypothesis)}
\end{aligned}$$

Set L' seems like a suitable candidate for L_{i+1} , however we might have $|L'| > t$. We call the following linear programming $\text{Pruning}(L')$. Let $L' = \{[(x^1, y^1), \lambda_1], \dots, [(x^k, y^k), \lambda_k]\}$. The variables of $\text{Pruning}(L')$ is a scalar variable θ_j for each node j in L' .

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

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

$$\theta \geq 0 \quad (19)$$

Notice that $\theta = \lambda$ is in fact a feasible solution to $\text{Pruning}(L')$. Let θ^* be the optimal vertex point solution to this LP. Since the problem is in \mathbb{R}^k , θ^* has to satisfy k 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 $j = 1, \dots, k$ 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 λ is feasible for $\text{Pruning}(L')$, we have $\sum_{j=1}^k \theta_j^* \geq \sum_{j=1}^k \lambda_j \geq \frac{1}{g^{t+1}}$. \square

The leaves of the FDT tree, or L_t have the property that every solution (x, y) in L_t has $x \in \{0, 1\}^n$ and $(x, y) \in \mathcal{D}(P)$. Our goal is now to replace these leaf solutions in $\mathcal{D}(P)$ with solutions in S . To this end, we need to reduce some variables with value of 1 into variables of value 0 and obtain feasibility in the meantime. We introduce a procedure that given a point $(x, y) \in \mathcal{D}(P)$ with $x \in \{0, 1\}^n$ outputs a point $z \leq x$ with $z \in \{0, 1\}^n$ such that (z, w) is in S for some vector $w \in \mathbb{R}^p$.

The algorithm “fixes” the variables iteratively, from x_1 to 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 $i = 1, \dots, \ell$. Now consider the following linear programming. The variables of this LP are the $z \in \mathbb{R}^n$ variables and $w \in \mathbb{R}^p$.

$$\text{DominantToFeasible}(x^{(\ell)}) \quad \min \quad z_{\ell+1} \quad (20)$$

$$\text{s.t.} \quad Az + Gw \geq b \quad (21)$$

$$z_j = x_j^{(\ell)} \quad j = 1, \dots, \ell \quad (22)$$

$$z_j \leq x_j^{(\ell)} \quad j = \ell + 1, \dots, n \quad (23)$$

$$z \geq 0 \quad (24)$$

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

This procedure is summarized in Algorithm 1.

Algorithm 1: The DomToIP algorithm

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

```

1  $x^{(0)} \leftarrow x$ 
2 for  $\ell = 0$  to  $t - 1$  do
3    $x^{(\ell+1)} \leftarrow x^{(\ell)}$ 
4    $\eta \leftarrow$  optimal value of DominantToFeasbile( $x^{(\ell)}$ )
5    $y^{(\ell+1)} \leftarrow$  optimal solution for  $w$  variables in DominantToFeasbile( $x^{(\ell)}$ )
6   if  $\eta = 0$  then
7      $x_{\ell+1}^{(\ell+1)} \leftarrow 0$ 
8   else
9      $x_{\ell+1}^{(\ell+1)} \leftarrow 1$ 
10  end
11 end

```

The following lemma proves Theorem 1.

Lemma 4. *Given $(x, y) \in \mathcal{D}(P)$ Algorithm 1 correctly finds $(x^{(n)}, y^{(n)}) \in S$ in polynomial time.*

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

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

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

Now assume $\text{IP}^{(\ell)}$ is non-empty for some $\ell \in \{0, \dots, t-2\}$. Since $\text{IP}^{(\ell)} \subseteq \text{LP}^{(\ell)}$ we have $\text{LP}^{(\ell)} \neq \emptyset$ 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 $z^* \leq x^{(\ell+1)}$, we have $(z^*, w^*) \in \text{LP}^{(\ell+1)}$. Also, $(z^*, w^*) \in P$. By Theorem 1 there exists

$(\tilde{z}^i, \tilde{w}^i) \in S$ and $\theta_i \geq 0$ for $i = 1, \dots, k$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz^*$. We have

$$\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz^* \leq gx^{(\ell+1)} \quad (25)$$

So for $j \in \{1, \dots, n\}$ where $x_j^{(\ell+1)} = 0$, we have $z_j^i = 0$ for $i = 1, \dots, k$. Hence, $z^i \leq x^{(\ell+1)}$ for $i = 1, \dots, k$. Hence, there exists $(z, w) \in S$ such that $z \leq x^{(\ell+1)}$. We claim that $(z, w) \in \text{IP}^{(\ell+1)}$. If $(z, w) \notin \text{IP}^{(\ell+1)}$ 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 Algorithm 1. Notice that $z \leq x^{(\ell+1)} \leq x^{(j)}$. We have $x_j^{(j)} = 1$ which implies when we solved $\text{DominantToFeasible}(x^{(j-1)})$ the optimal value was strictly larger than zero. However, (z, w) is a feasible solution to $\text{DominantToFeasible}(x^{(j-1)})$ and gives an objective value of 0. This is a contradiction, so $(z, w) \in \text{IP}^{(\ell+1)}$.

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

Now consider $(x^{(t)}, y^{(t)})$. Let $(z, y^{(t)})$ be the optimal solution to $\text{LP}^{(t-1)}$. 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 $(z', w') \in \text{IP}^{(t-1)}$. We show that $x^{(t)} = z'$. Observe that for $j = 1, \dots, n-1$ 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 $(z', w') \in \text{LP}^{(t-1)}$ has objective value of 0 for $\text{DominantToFeasible}(x^{(t-1)})$, this is a contradiction to (z, w) being the optimal solution. \square

This completes the FDT algorithm.

Algorithm 2: Fractional Decomposition Tree Algorithm

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

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

```

1  $L^0 \leftarrow [(x^*, y^*), 1]$ 
2 for  $i = 1$  to  $t$  do
3    $L' \leftarrow \emptyset$ 
4   for  $[(x, y), \lambda] \in L^i$  do
5     Apply Lemma 2 to obtain  $[(\hat{x}^0, \hat{y}^0), \gamma_0]$  and  $[(\hat{x}^1, \hat{y}^1), \gamma_1]$ 
6      $L' \leftarrow L' \cup \{[(\hat{x}^0, \hat{y}^0), \gamma_0]\} \cup \{[(\hat{x}^1, \hat{y}^1), \gamma_1]\}$ 
7   end
8   Apply Lemma 3 to  $L'$  to obtain  $L^{i+1}$ .
9 end
10 for  $[(x, y), \lambda] \in L^t$  do
11   Apply Algorithm 1 to  $(x, y)$  to obtain  $(z, w) \in S$ 
12    $F \leftarrow F \cup \{[(z, w), \lambda]\}$ 
13 end
14 return  $F$ 

```

3 FDT for 2EC

In Section 2 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 $2EC(G) = \{x \in \mathbb{R}^E : x(\delta(U)) \geq 2 \text{ for } \emptyset \subset U \subset V\}$. Notice that $\mathcal{D}(2EC(G)) \cap [0, 2]^E = 2EC(G)$, since 2EC is a covering problem. We want to 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, \dots, F_k such that $k \leq 2|V|$, $\sum_{i=1}^k \lambda_i \chi^{F_i} \leq Cx^*$, and $\sum_{i=1}^k \lambda_i = 1$. Moreover, $C \leq g_{2EC}^{2|V|}$, where g_{2EC} is the integrality gap of the 2-edge-connected multigraph problem with respect to formulation in (6)*

We do not know the exact value for g_{2EC} , but we know $\frac{6}{5} \leq g_{2EC} \leq \frac{3}{2}$ [?]. 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 show later in Section ?? that in practice the constant C for the FDT algorithm for 2EC is much better than $\frac{3}{2}$.

The FDT algorithm for 2EC is very similar to the one for binary MIPs, but there are some differences as well. First, we need a branching lemma. Observe that the following branching lemma is essentially a translation of Lemma 1 for 0,1,2 problems except for one additional clause.

Lemma 5. *Given $x \in 2EC(G)$, 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*

$$(i) \quad \gamma_0 + \gamma_1 + \gamma_2 \geq \frac{1}{g_{2EC}},$$

$$(ii) \quad x^0, x^1, \text{ and } x^2 \text{ are in } 2EC(G),$$

$$(iii) \quad x_e^0 = 0, x_e^1 = 1, \text{ and } x_e^2 = 2,$$

$$(iv) \quad \gamma_0 x^0 + \gamma_1 x^1 + \gamma_2 x^2 \leq x,$$

$$(v) \quad \text{for } f \in E \text{ with } x_f \geq 1, \text{ we have } x_f^j \geq 1 \text{ for } j = 0, 1, 2.$$

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

$$\max \quad \sum_{j=0,1,2} \lambda_j \tag{26}$$

$$\text{s.t.} \quad x^j(\delta(U)) \geq 2\lambda_j \quad \text{for } \emptyset \subset U \subset V, \text{ and } j = 0, 1, 2 \tag{27}$$

$$0 \leq x^j \leq 2\lambda_j \quad \text{for } j = 0, 1, 2 \tag{28}$$

$$x_e^j = j \quad \text{for } j = 0, 1, 2 \tag{29}$$

$$x_f^j \geq j \quad \text{for } f \in E \text{ where } x_f \geq 1, \text{ and } j = 0, 1, 2 \tag{30}$$

$$x^0 + x^1 + x^2 \leq x \tag{31}$$

$$\lambda_0, \lambda_1, \lambda_2 \geq 0 \tag{32}$$

Let x^j, γ_j for $j = 0, 1, 2$ be an optimal 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. In order to show that (i) is also satisfied we prove the following claim.

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

Proof. Suppose for contradiction $\sum_{j=0,1,2} \gamma_j = \frac{1}{g_{2EC}} - \epsilon$ 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 $i = 1, \dots, k$ 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, \dots, F_k of G using F'_1, \dots, 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 $\chi_h^{F'_i} = 0$, this means that for any edge $p \in P_f$ such that $p \neq h$, $\chi_p^{F'_i} = 2$. In this case, let $\chi_f^{F_i} = 2$, i.e. add two copies of f to F_i . If there is an edge $h \in P_f$ with $\chi_h^{F'_i} = 1$, let $\chi_f^{F_i} = 1$, i.e. add one copy of f to F_i . If for all edge $h \in P_f$, we have $\chi_h^{F'_i} = 2$, then let $\chi_f^{F_i} = 2$. For $f \in E$ with $x_f < 1$ we have

$$\sum_{i=1}^k \theta_i \chi_f^{F_i} = \sum_{i=1}^k \theta_i \chi_f^{F'_i} \leq g_{2\text{EC}} x'_f = g_{2\text{EC}} x_f. \quad (33)$$

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.

$$\sum_{i=1}^k \theta_i \chi_f^{F_i} \leq \sum_{i=1}^k \theta_i \frac{\sum_{h \in P_f} \chi_h^{F'_i}}{\lceil \frac{2}{\epsilon} \rceil - 1} \quad (34)$$

$$= \frac{\sum_{h \in P_f} \sum_{i=1}^k \theta_i \chi_h^{F'_i}}{\lceil \frac{2}{\epsilon} \rceil - 1} \quad (35)$$

$$\leq \frac{\sum_{h \in P_f} g_{2\text{EC}} x'_h}{\lceil \frac{2}{\epsilon} \rceil - 1} \quad (36)$$

$$= \frac{\sum_{h \in P_f} g_{2\text{EC}} x'_h}{\lceil \frac{2}{\epsilon} \rceil - 1} \quad (37)$$

$$= \frac{\lceil \frac{2}{\epsilon} \rceil}{\lceil \frac{2}{\epsilon} \rceil - 1} g_{2\text{EC}} x_f \quad (38)$$

Therefore, we have

$$x' \geq \sum_{i \in [k]: \chi_e^{F_i} = 1} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\text{EC}} \lceil \frac{2}{\epsilon} \rceil} \chi^{F_i} + \sum_{i \in [k]: \chi_e^{F_i} = 2} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\text{EC}} \lceil \frac{2}{\epsilon} \rceil} \chi^{F_i}. \quad (39)$$

Let $x^j = \sum_{i \in [k]: \chi_e^{F_i} = j} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\text{EC}} \lceil \frac{2}{\epsilon} \rceil} \chi^{F_i}$ and $\theta_j = \sum_{i \in [k]: \chi_e^{F_i} = j} \frac{\theta_i (\lceil \frac{2}{\epsilon} \rceil - 1)}{g_{2\text{EC}} \lceil \frac{2}{\epsilon} \rceil}$ 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

$\sum_{j=0,1,2} \theta_j = \frac{\lceil \frac{2}{\epsilon} \rceil - 1}{g_{2EC} \lceil \frac{2}{\epsilon} \rceil}$. By assumption, we have $\frac{\lceil \frac{2}{\epsilon} \rceil - 1}{g_{2EC} \lceil \frac{2}{\epsilon} \rceil} \leq \frac{1}{g_{2EC}} - \epsilon$. However, this means $2 \leq 1$ which is a contradiction. \square

This concludes the proof. \square

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 $i = 1, \dots, |\sup(x^*)|$ be collections of pairs of feasible points in $2EC(G)$ together with their multipliers. Let $t = |\sup(x^*)|$ and assume without loss of generality that $\sup(x^*) = \{e_1, \dots, e_t\}$.

Lemma 6. *The FDT algorithm for 2EC in polynomial time produces sets L_0, \dots, L_t of pairs $x \in 2EC(G)$ 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 $j = 1, \dots, i$,
- b. $\sum_{(x,\lambda) \in L_i} \lambda \geq \frac{1}{g_{2EC}^i}$,
- c. $\sum_{(x,\lambda) \in L_i} \lambda x \leq x^*$,
- d. $|L_i| \leq t$.

Proof. We proceed by induction on the 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 $i = 1, \dots, t$ 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 3 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 $e \in \{e_1, \dots, e_{i-1}\}$ if $x_e^\ell = 0$, then by property (iv) in Lemma 5 we have $x_e^{\ell j} = 0$ for $j = 0, 1, 2$. Otherwise by induction we have $x_e^\ell \geq 1$ in which case property (v) in Lemma 5 ensures that $x_e^{\ell j} \geq 1$ for $j = 0, 1, 2$. Also, $x_{e_i}^{\ell j} = j$, so $x_{e_i}^{\ell j} = 0$ or $x_{e_i}^{\ell j} \geq 1$ for $j = 0, 1, 2$.

If $|L'| \geq t$ we let $L_i = L'$, otherwise apply Pruning(L') to obtain L_i . \square

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 $\lfloor x \rfloor \in 2EC(G)$.*

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 $x \in 2EC(G)$ 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. \square

4 TO DO

- Implement FDT on SND 0,1,2 and point-to-point SND 0,1,2
- Implement of TAP
- Write paper for IPCO

References

- [CV04] 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.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.
- [Sch03] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.