Unsplittable Flow Problem on Paths and Trees: Closing the LP Relaxation Integrality Gap

Adam Jozefiak* Yuchong Pan†

Abstract

This is a survey of a series of papers which address the existence of LP relaxations for the Unsplittable Flow Problem (UFP) on paths and trees which have small integrality gaps. On paths and trees UFP is nice in the sense that each request, if routed, must be routed on a unique path, allowing for the problem to be modelled by a simple integer program and additionally a simple natural LP relaxation. We begin our discussion with [4] where Chekuri et al. demonstrate that with the no bottleneck assumption the natural LP relaxation attains an $O(1)$ integrality gap on UFP instances on the path and tree. However, removing this no bottle neck assumption results in an $\Omega(n)$ integrality gap for the natural LP relaxation for UFP instances on the path and tree [3], where n is the number of vertices in the graph in the instance. In response to this lower bound, we discuss the work of Chekuri et al. who present LP relaxations for UFP instances on the path without the no bottleneck assumption that attain an integrality gap of $O(\log(n))$. Chekuri et al.ś new LP relaxations utilize rank constraints on subsets of big requests, requests whose demand is at least a $\frac{3}{4}$ of one of the edge capacities along the unique path of the request. Chekuri et al. are unable to generalize their results to UFP instances on trees, however, Friggstad and Gao in [7] present an LP relaxation for UFP instances on trees that is a natural generalization of [6]'s new rank constraint based LP relaxation. Friggstad and Gao's LP relaxation is shown to attain an integrality gap of $O(\log(n) \cdot \min\{\log(n), \log(k)\})$, matching an approximation factor for a greedy combinatorial algorithm for UFP instances on trees presented in [6]. Finally, we observe that Anagnostopoulos et al. in [1] present a new LP relaxation for UFP on paths which embeds a dynamic program into the LP and is shown to have an $O(1)$ integrality gap.

1 Introduction

In the general Unsplittable Flow Problem (UFP), we are given a graph $G = (V, E)$, with a non-negative edge capacity $c_e \geq 0$ for each $e \in E$. Throughout this paper, we will use $n = |V|$ and $m = |E|$. In addition, we are given a set of k requests $\mathcal{R} = \{R_1, \dots, R_k\}$, each of which consists of a pair of vertices $(s_i, t_i) \in V \times V$, a nonnegative demand $d_i \geq 0$, and a nonnegative weight $w_i \geq 0$. Each request $R_i = ((s_i, t_i), d_i, w_i)$ asks to route d_i units of flow on a *single* path from s_i to t_i, producing w_i units of weight. We say that a subset $\mathcal{S} \subseteq \mathcal{R}$ is *routable* if we can simultaneously route all requests in $\mathcal{S}$ without violating the capacity constraints of edges. The goal of UFP is to find a routable subset $\mathcal{S}$ of requests that maximizes the total weight $\sum_{R_i \in S} w_i$.

In this survey, we chiefly focus on two special cases of UFP, Unsplittable Flow Problem on Paths (UFP-Path) and Unsplittable Flow Problem on Trees (UFP-Tree). In these two special cases, the given graph $G = (V, E)$ is guaranteed to be a tree or a path, respectively, so there

*adam.jozefiak@alumni.ubc.ca, University of British Columbia.
†yuchong.pan@alumni.ubc.ca, University of British Columbia.

exists a unique path between s_i and t_i for each request. We will use P_i to denote the unique path between s_i and t_i in G for each request $R_i \in \mathcal{R}$. Therefore, a routable subset $\mathcal{S}$ of requests can be characterized by $\sum_{R_i \in \mathcal{S}, e \in P_i} d_i \leq c_e$ for all $e \in E$.

Compared to the general UFP problem, UFP-Path and UFP-Tree eliminate the hardness of choosing which path between s_i and t_i to route each request because of the uniqueness of the path between s_i and t_i. Nevertheless, UFP-Path and UFP-Tree are still difficult in selecting which requests to route. For example, the instance of UFP-Path that only consists of one edge is equivalent to the Knapsack problem as it maximizes the total weight given a fixed size. The reduction of Knapsack to UFP-Path shows that UFP-Path and UFP-Tree are NP-hard.

Given an instance $(G, \mathcal{R})$ of UFP-Tree or UFP-Path, one can easily formulate an integer program for the instance. Let x_i represent whether or not to route each request $R_i \in \mathcal{R}$. Therefore, an integer programming formulation is given as follows. We call this integer program (UFP-IP).

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} w_i x_i & & \text{(UFP-IP)} \\ \text{s.t.} \quad & \sum_{R_i \in \mathcal{R}, e \in P_i} d_i x_i \leq c_e, & \forall e \in E, & \\ & x_i \in \{0, 1\}, & \forall R_i \in \mathcal{R}. & \end{aligned}$$

A natural way to obtain a linear programming relaxation is to relax the constraints $x_i \in \{0, 1\}$ to $x_i \in [0, 1]$ for all $R_i \in \mathcal{R}$. Hence, UFP-IP leads to a natural linear programming relaxation which we call (UFP-LP).

$$\begin{aligned} \max \quad & \sum_{i=1}^{k} w_i x_i & & \text{(UFP-LP)} \\ \text{s.t.} \quad & \sum_{R_i \in \mathcal{R}, e \in P_i} d_i x_i \leq c_e, & \forall e \in E, & \\ & x_i \in [0, 1], & \forall R_i \in \mathcal{R}. & \end{aligned}$$

In this survey, we present a variety of linear programming relaxations for UFP-Path and for UFP-Tree, which can be solved in polynomial time, with the goal of minimizing the integrality gap between the linear programming relaxation and (UFP-IP). The *integrality gap* is an important concept in approximation algorithms that measures the gap between the optimal (fractional) solution to a linear programming relaxation and the optimal integral solution to the original problem. For UFP-Path and UFP-Tree, the *integrality gap* of a linear programming formuation $\max \left\{ c^\top x : Ax \leq b, x \geq 0 \right\}$ is defined to be

$$\frac{\max \left\{ c^\top x : Ax \leq b, x \geq 0 \right\}}{\text{OPT}},$$

where OPT denotes the optimal solution to (UFP-IP).

2 Natural LP Relaxation with No Bottleneck Assumption

We say that a UFP instance satisfies the *no bottleneck assumption (NBA)* if

$$\max_{R_i \in \mathcal{R}} d_i \leq \min_{e \in E} c_e.$$

The no bottleneck assumption has been shown to be a strong assumption for UFP-Tree. Chekuri, Mydlarz, and Bruce prove that the natural linear programming relaxation (UFP-LP) has an $O(1)$ integrality gap with NBA [4].

First, Chekuri, Mydlarz, and Bruce consider the special case of UFP-Tree where all demands are 1 (called *unit demands*), i.e., $d_i = 1$ for all $R_i \in \mathcal{R}$ [4]. This special case is also named the *Integer Multicommodity Flow Problem.* Chekuri, Mydlarz, and Bruce prove the following theorem for the unit demand case of UFP-Tree [4]:

Theorem 2.1. *If (*UFP-LP*) has a feasible solution x of value O, then it has a feasible integral solution z of value at least $\frac{O}{4}$. Moreover, given such an x, we may compute such a z in polynomial time.*

Theorem 2.1 directly implies that the integrality gap of UFP-Tree with unit demands is at most 4, and is hence $O(1)$. To prove Theorem 2.1, Chekuri, Mydlarz, and Bruce prove a coincident result, presented as follows [4]:

Theorem 2.2. *Let x be a feasible solution to (*UFP-LP*) with unit demands. Let $k \in \mathbb{Z}$ such that kx is integral. Then there exist feasible integral solutions $z_1, \ldots, z_{4k}$ such that $kx \leq \sum_{i=1}^{4k} z_i$.*

We believe that it is interesting to note that the proof of Theorem 2.2 is closely related to the *Binned Tree Coloring Problem.* In a Binned Tree Coloring Problem, we are given a tree T, rooted at a fixed leaf node v^*, with an edge capacity $c_e \geq 0$ for each edge $e \in E$. Additionally, we are given an integer $k \in \mathbb{Z}$ and a multiset of requests $\mathcal{R}$, defined as in UFP. Without loss of generality, we may assume that s_i and t_i in each request $R_i \in \mathcal{R}$ are leaves in the tree. This can be done by adding new supply edges. We assume that the number of pairs (s_i, t_i) in the *fundamental cut* of e is at most kc_e, where the *fundamental cut* of an edge e is defined to be the set of all edges (u, v) such that u and v lie in different connected components in $T - e$. We further assume that each leaf $v \neq v^*$ partitions requests incident to v into at most $n_v \leq c_{(v,p(v))}$ bins $B_1(v), \ldots, B_{n_v}(v)$ with $|B_i(v)| \in [1, 2k)$ for each i, where $p(v)$ denotes the parent of v in T. The goal of the Binned Tree Coloring Problem is to find a coloring of requests such that each color class $\mathcal{R}_i$ of requests is routable and such that the edges in $B_i(v)$ all have different colors for each leaf v and for each $i \in \{1, \ldots, n_v\}$.

We now proceed to present the result of Chekuri, Mydlarz, and Bruce for the general demand case of UFP-Tree with NBA [4]:

Theorem 2.3. *The integrality gap of the natural linear programming relaxation (*UFP-LP*) is at most* 48 *for the general demand case of* UFP-Tree *with the no bottleneck assumption. Moreover, we may find an integral solution of value at least $\frac{1}{48}$ times of the value of the optimal solution to (*UFP-LP*) in polynomial time.*

To prove Theorem 2.3, Chekuri, Mydlarz, and Bruce appeal to *column-restricted packing integer programs (CPIP)*, which we define below [4]. Let $A \in \{0,1\}^{m \times n}$ be a $\{0,1\}$ matrix with m rows and n columns. For $d \in \mathbb{R}^n_+$, we define $A[d]$ to be the matrix obtained by multiplying each component in the i^{th} column of A by d_i. Then each CPIP instance is of the form $\max\{wx : Ax \leq b, x \in \{0,1\}^n\}$. In order to state their theorem for CPIP instances, we first introduce several required definitions:

Definition 2.1. For a convex body $P \subseteq \mathbb{R}^n$, the *integer hull* of P, denoted by P_I, is the convex hull of all integral vectors in P.

Definition 2.2. For a convex body $P \subseteq \mathbb{R}^n$, the *integrality gap* of the optimization problem $\gamma = \max\{wx : x \in P\}$ is defined to be the ratio of the optimal (fractional) value γ to the optimization problem to the optimal value of an integral solution, i.e.,

$$\frac{\gamma}{\max\{wx : x \in P_I\}}.$$

Definition 2.3. We say that a set $W \subseteq \mathbb{Z}^n$ of vectors is *closed* if for all $w \in W$, the vector w' obtained by replacing some of the components of W with 0 is also in W.

Definition 2.4. For a matrix A and a closed set W of vectors, we denote by $\mathcal{P}(A, w)$ the class of problems of the form $\max\{wx : Ax \leq b, x \in [0,1]^n\}$ for some $w \in W$ and $b \in \mathbb{Z}_+^m$. In addition, we denote by $\mathcal{P}^{\text{dem}}(A, W)$ the class of problems of the form $\max\{wx : A[d]x \leq b, x \in [0,1]^n\}$ for some $w \in W$ and $b, d \in \mathbb{Z}_+^m$ with $d_{\max} \leq b_{\min}$, where $d_{\max}$ and $b_{\min}$ denote the maximum and minimum components of d and of b, respectively.

Definition 2.5. The *integrality gap* of a class $\mathcal{P}$ of problems is defined to be the supremum of the integrality gaps of individual problems in $\mathcal{P}$.

Now, we are able to state the theorem of Chekuri, Mydlarz, and Bruce for CPIP instances [4]. Chekuri, Mydlarz, and Bruce also extend the theorem to nonnegative matrices [4].

Theorem 2.4. *Let A be a $\{0,1\}$ matrix and W be a closed set of vectors. If the integrality gap of $\mathcal{P}(A, W)$ is at most Γ, then the integrality gap of $\mathcal{P}^{dem}(A, W)$ is at most $11.542\Gamma \leq 12\Gamma$.*

Theorem 2.5. *Let A be a nonnegative matrix and W be a closed set of vectors. If the integrality gap of $\mathcal{P}(A, W)$ is at most Γ, then the integrality gap of $\mathcal{P}^{dem}(A, W)$ is at most $11.542\Gamma \leq 12\Gamma$.*

By setting $d_i = 1$ for each request $R_i \in \mathcal{R}$, we obtain from (UFP-LP) the following natural linear programming relaxation for the unit demand case of UFP-Tree, which we denote by (UFP-LP-Unit).

$$\begin{array}{llr}
\max & \displaystyle\sum_{i=1}^{k} w_i x_i & \text{(UFP-LP-Unit)} \\
\text{s.t.} & \displaystyle\sum_{R_i \in \mathcal{R}, e \in P_i} x_i \leq c_e, & \forall e \in E, \\
& x_i \in [0,1], & \forall R_i \in \mathcal{R}.
\end{array}$$

By comparing (UFP-LP) and (UFP-LP-Unit), one can easily see that if $\mathcal{P}(A, W)$ is the class of UFP-Tree instances with unit demands, then $\mathcal{P}^{\text{dem}}(A, W)$ is the class of UFP-Tree instances with general demands. By Theorem 2.1, the integrality gap of UFP-Tree instances with unit demands is at most 4. Hence, Theorem 2.5 implies that the integrality gap of UFP-Tree instances with general demands is at most 48.

3 Integrality Gap Lower Bound without No Bottleneck Assumption

The no bottleneck assumption is essential to the results in [4]. Chakrabarti, Chekuri, Gupta, and Kumar present a canonical example which shows the following theorem [3]:

Theorem 3.1. *The lower bound of the integrality gap of the general* UFP-Tree *or* UFP-Path *problems without NBA is $\Omega(n)$.*

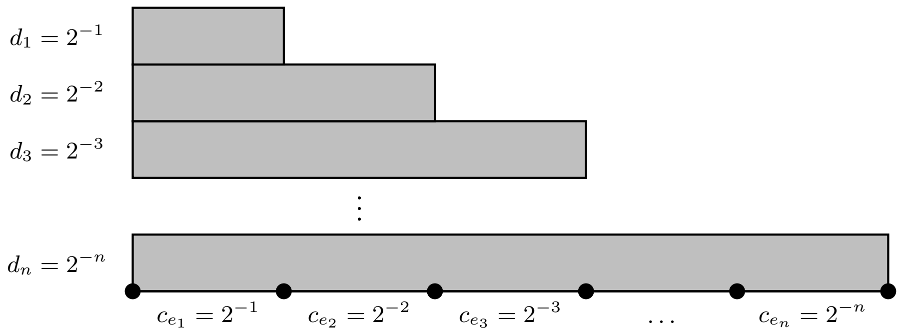

Figure 1: A canonical example which shows that the lower bound of the integrality gap of UFP-Tree without NBA is $\Omega(n)$.

Proof. Let $G = (V, E)$ be the path graph on $n + 1$ vertices with the vertex set $V = \{v_0, \ldots, v_n\}$ and the edge set $E = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n)\}$. Let the edge capacity of edge (v_{i-1}, v_i) be $c_{(v_{i-1}, v_i)} = 2^{-i}$ for each $i = 1, \ldots, n$. Let $\mathcal{R} = \{R_1, \ldots, R_n\}$ be n requests, each R_i of which asks to route $d_i = 2^{-i}$ unit of flow from v_0 to v_i, producing $w_i = 1$ unit of weight. The example is illustrated in Fig. 1.

First, we show that the value of any integral optimal solution equals 1. Let $R_i = \left((v_0, v_i), 2^{-i}, 1\right) \in \mathcal{R}$ be a request. Recall that the path from v_0 to v_i consists of edges $(v_0, v_1), \ldots, (v_{i-1}, v_i)$, with edge capacities $2^{-1}, 2^{-2}, \ldots, 2^{-i}$, respectively. Since $2^{-i} \le 2^{-j}$ for each $j = 1, \ldots, i$, then we route R_i to produce 1 unit of weight. Let $R_j = \left((v_0, v_j), 2^{-j}, 1\right) \in \mathcal{R}$ with $j \neq i$. If $j > i$, then the path from v_0 to v_j contains edge (v_{i-1}, v_i), whose capacity 2^{-i} has been used up by R_i, so R_j cannot be routed. If $j < i$, then the path from v_0 to v_j contains edge (v_{j-1}, v_j) whose capacity is 2^{-j}. Since R_i has used 2^{-i} units of capacity, the residual capacity is $2^{-j} - 2^{-i} < 2^{-j} = d_j$, so R_j cannot be routed. This implies that any optimal integral solution consists of exactly one request. Since each request produces 1 unit of weight, then the value of any optimal integral solution equals 1.

Second, we present a feasible fractional solution to (UFP-LP) that gives value $\frac{n}{2}$. Let $x_i = \frac{1}{2} \in [0, 1]$ for each $R_i \in \mathcal{R}$. For each $e = (v_{j-1}, v_j) \in E$, we have

$$\begin{aligned}\sum_{R_i \in \mathcal{R}, e = (v_{j-1}, v_j) \in P_i} d_i x_i &= \sum_{i=j}^{n} 2^{-i} \cdot \frac{1}{2} = \sum_{i=j}^{n} 2^{-i-1} = 2^{-j-1} \cdot \frac{1 - \left(\frac{1}{2}\right)^{n-j+1}}{1 - \frac{1}{2}} \\ &= 2^{-j} \cdot \left(1 - 2^{j-n-1}\right) = 2^{-j} - 2^{-n-1} < 2^{-j} = c_e.\end{aligned}$$

This implies that the capacity constraint for each edge is satisfied and therefore that $x_i = \frac{1}{2}, \forall R_i \in \mathcal{R}$ is a feasible solution to (UFP-LP). Thus, the total weight produced is

$$\sum_{i=1}^{k} w_i x_i = \sum_{i=1}^{n} 1 \cdot \frac{1}{2} = \frac{n}{2}.$$

Therefore, the value of any optimal (fractional) solution to (UFP-LP) is at least $\frac{n}{2}$. This completes the proof that the integrality gap of UFP-LP for the general UFP-Tree without NBA is lower bounded by $\frac{\frac{n}{2}}{1} = \frac{n}{2} = \Omega(n)$. □

The $\Omega(n)$ lower bound of the integrality gap of the natural linear programming relaxation (UFP-LP) implies that (UFP-LP) is not tight for UFP-Path and for UFP-Tree without NBA. This strongly indicates that strengthenings of the natural linear programming relaxation (UFP-LP) are desired, which we will discuss in the following sections.

4 LP Relaxations with Rank Constraints for UFP-Path without NBA

Chekuri, Ene, and Korula describe two new LP relaxations for UFP-Path without NBA that both have a $O(\log(n))$ integrality gap [6]. Prior to [6], there was no known LP relaxation for UFP-Path with an integrality gap of $o(n)$. Chekuri et al's first LP relaxation which we call (UFP-RankLP) contains an exponential number of constraints; however, the authors are only able to discover an approximate separation oracle. Their second LP relaxation which we call (UFP-CompactRankLP) is a compact relaxation containing a polynomially bounded subset of the first LP's constraints; however, (UFP-RankLP) is still strictly tighter than (UFP-CompactRankLP). In particular, Chekuri et al show that any feasible solution to (UFP-CompactRankLP) when scaled by a constant factor is feasible for (UFP-RankLP).

The canonical example presented in [3] and discussed in the previous section of this survey demonstrates that the natural LP relaxation has an $\Omega(n)$ integrality gap, and hence is not tight enough for UFP-Path and UFP-Tree without NBA. A common technique in order to strengthen and tighen LP relaxations, and in turn to reduce integrality gaps, is to add valid constraints to the LP relaxation. In this spirit, Chekuri et al add new rank constraints to the natural LP relaxation in order to obtain $O(\log(n))$ integrality gaps.

First we define a rank constraint in general. Given a set of requests $\mathcal{S} \subseteq \mathcal{R}$, we define rank$(\mathcal{S})$ to be the maximum number of requests in $\mathcal{S}$ that can be routed simultaneously. Then the corresponding rank constraint given by

$$\sum_{R_i \in \mathcal{S}} x_i \leq rank(\mathcal{S}).$$

We proceed to describe Chekuri et al's new LP relaxations for UFP-Path without the NBA and state several of the authors' results. Before providing some intuition behind Chekuri et al's LP relaxations we provide some definitions. Let $R_i \in \mathcal{R}$. We define the *bottleneck edge* of request R_i to be the edge in P_i, the unique s_i-t_i path in G, with the minimal capacity. If there are multiple edges in P_i with the minimal capacity, the leftmost edge is selected to be the bottleneck edge. Small requests are defined as

$$\mathcal{S} = \left\{ R_i \in \mathcal{R} : e \text{ is the bottleneck of } R_i \text{ and } d_i \leq \frac{3}{4} c_e \right\}.$$

Then big requests are defined as

$$\mathcal{B} = \mathcal{R} \setminus \mathcal{S}.$$

To succinctly summarize Chekuri et al's new insights, their new LP relaxations use rank constraints on big requests, which in fact are what make UFP-Path and UFP-Tree difficult without NBA in the canonical example as every request passes through an edge for which its capacity equals the request's demand. Therefore, Chekuri et al maintain the constraints $\sum_{R_i \in \mathcal{R}, e \in P_i} x_i \leq c_e$ for each edge $e \in E$ to deal with small requests and additionally create new rank constraints for subsets of big requests.

We provide a few more definitions. The following subsets of $\mathcal{B}$ are defined:

$$\mathcal{B}_e = \{ R_i \in \mathcal{B} : e \in P_i \}, \qquad \forall e \in E.$$

Then using rank constraints on the collections of sets $\{\mathcal{B}_e : e \in E\}$, Chekuri et al provide the following LP relaxation which we will call (UFP-RankLP) (we note that Chekuri et al call the LP (UFP-LP)):

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{k} w_i x_i & & \text{(UFP-RankLP)}\\
\text{s.t.} \quad & \sum_{R_i \in \mathcal{R}, e \in P_i} d_i x_i \le c_e, & & \forall e \in E,\\
& \sum_{R_i \in B} x_i \le \operatorname{rank}(B), & & \forall e \in E, \forall B \subseteq \mathcal{B}_e,\\
& x_i \in [0,1], & & \forall R_i \in \mathcal{R}.
\end{aligned}
$$

In a previous paper [5], Chekuri et al showed that (UFP-RankLP) has an approximate separation oracle that allows (UFP-RankLP) to be solved to within a constant factor in polynomial time. The authors additionally discovered that this approximate separation oracle implicitly defines a compact relaxation which includes only a polynomially bounded subset of rank constraints. Chekuri et al show that this new compact LP relaxation has a $O(\log(n))$ integrality gap for UFP-Path without NBA and additionally for (UFP-RankLP). Before presenting the compact LP relaxation, which we will call (UFP-CompactRankLP), we present some necessary definitions. Let

- $\mathcal{B}_{\text{left}}(e) = \{R_i \in \mathcal{B}_e : \text{the bottleneck of } R_i \text{ is } e \text{ or is to the left of } e\}$;
- $\mathcal{B}_{\text{right}}(e) = \{R_i \in \mathcal{B}_e : \text{ the bottleneck of } R_i \text{ is } e \text{ or is to the right of } e\}$.

Additionally Chekuri et al define *sets of blocking requests* (which are generalized in Friggstad and Gao's paper [7]). We say that for all $R_i, R_j \in \mathcal{B}_e$, R_j *blocks* R_i if $d_j > d_i$ and if R_i and R_j are not simultaneously routable. Then the *blocking sets* are defined as follows:

$$
\begin{aligned}
\text{LeftBlock}(e,i) &= \{R_i\} \cup \{R_j \in \mathcal{B}_{\text{left}}(e) : R_j \text{ blocks } R_i\}, & \forall e \in E, \forall R_i \in \mathcal{B}_{left}(e),\\
\text{RightBlock}(e,i) &= \{R_i\} \cup \{R_j \in \mathcal{B}_{\text{Right}}(e) : R_j \text{ blocks } R_i\}, & \forall e \in E, \forall R_i \in \mathcal{B}_{right}(e).
\end{aligned}
$$

Additionally, Chekuri et al prove that the following theorem:

Theorem 4.1. *1.* $\operatorname{rank}(\text{LeftBlock}(e,i)) \le 1$ *for each* $e \in E$ *and each* $R_i \in \mathcal{B}_{left}(e)$*;*

2. $\operatorname{rank}(\text{RightBlock}(e,i)) \le 1$ *for each* $e \in E$ *and each* $R_i \in \mathcal{B}_{right}(e)$*.*

With these new blocking sets the new (UFP-CompactRankLP) formulation is given as follows:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{k} w_i x_i & & \text{(UFP-CompactRankLP)}\\
\text{s.t.} \quad & \sum_{R_i \in \mathcal{R}, e \in P_i} d_i x_i \le c_e, & & \forall e \in E,\\
& \sum_{R_j \in \text{LeftBlock}(e,i)} x_j \le 1, & & \forall e \in E, \forall R_i \in \mathcal{B}_{\text{left}}(e),\\
& \sum_{R_j \in \text{RightBlock}(e,i)} x_j \le 1, & & \forall e \in E, \forall R_i \in \mathcal{B}_{\text{right}}(e),\\
& x_i \in [0,1], & & \forall R_i \in \mathcal{R}.
\end{aligned}
$$

Checkuri et al then prove the following results:

Theorem 4.2. *Let x be any feasible solution to (*UFP-CompactRankLP*) on a given instance of* UFP-Path*. Then there is an absolute constant $\alpha \leq 18$ such that $\frac{x}{\alpha}$ is feasible for (*UFP-RankLP*) on the same instance.*

Theorem 4.3. *The LP relaxation Compact UFP-RankLP has a $O(\log(n))$ integrality gap for instances of UFP-Path.*

Corollary 4.1. *The LP relaation UFP-RankLP has $O(\log(n))$ integrality gap for instances of UFP-Path.*

We remark that Corollary 4.1 is a consequence of Theorem 4.3 and Theorem 4.2. Additionally, we remark that retaining only a subset of the rank constraints of (UFP-RankLP) results in at most a constant factor increase in the integrality gap, thus providing credence to the conjecture that more rank constraints may not be sufficient to further reduce the integrality gap.

Additionally, Chekuri et al begin their paper by providing an $O(\log(n) \min\{\log(n), \log(k)\})$ greedy approximation algorithm for UFP-Tree without NBA. The authors note that the new constraints in (UFP-RankLP) are constructed to follow the analysis of this greedy approximation algorithm on UFP-Tree. In spite of the fact that this greedy approximation algorithm and its analysis motivate the new constraints in (UFP-RankLP), the authors are unable to formulate and prove an LP relaxation for UFP-Tree without NBA has an $o(n)$ integrality gap. Consequently, the authors leave the following open questions:

- Can the rank based constraints in (UFP-RankLP) and (UFP-CompanctRankLP) be generalized for an LP relaxation for UFP-Tree with a $o(n)$ integrality gap?
- Is there any LP relaxation for UFP-Tree with an $o(n)$ integrality gap, not necessarily utilizing rank based constraints?
- Is there an LP relaxation for UFP-Path with an $o(\log(n))$ integrality gap? In particular, the authors believe that an $O(1)$ integrality gap is a likely result.

In fact, it is the case that all three open questions are unanswered by the next two papers in this survey.

5 Generalizing Rank Constraints to UFP-Tree without NBA

Friggstad and Gao generalize the rank constraints in [6]'s LP relaxations for UFP-Path instances to UFP-Tree instances [7]. Friggstad and Gao present an LP relaxation for UFP-Tree with polynomially many constraints and show that this LP relaxation attains an integrality gap of $O(\log(n) \min\{\log(n), \log(k)\})$. It is interesting to note that this integrality gap matches the approximation ratio of [6]'s greedy approximation algorithm on UFP-Tree instances. Furthermore, Chekuri et al utilize their greedy approximation algorithm to motivate their rank based LP relaxations for UFP-Path instances and prove bounds on their integrality gaps; however, they leave as an open question where their LP relaxations can be generalized to instances of UFP-Tree which attain an $o(n)$ integrality gap. In fact [7] is the first paper to demonstrate an LP relaxation for UFP-Tree without NBA that attains an $o(n)$ integrality gap.

Next we define the *blocking sets* utilized by Friggstad and Gao, along with a few details, and then present their LP relaxation attaining the desired integrality gap on UFP-Tree instances. For each request $R_i \in \mathcal{R}$, for each vertex $v \in V$ spanned by P_i, and for each endpoint $a \in \{s_i, t_i\}$, a *blocking set* $C(i, v, a) \subseteq \mathcal{R}$ is defined to include $\{R_i\}$ and all other requests R_j satisfying

- v is also spanned P_j;
- $d_j \geq d_i$;
- $d_i + d_j > c_e$ for some e on the path between a and v and in the span of P_j.

Additionally, Friggstad and Gao show that the following theorem, an analogous result for the blocking sets in [6]:

Theorem 5.1. *rank*$(C(i, v, a) = 1$ *for each* $R_i \in \mathcal{R}$*, for each* $v \in V$*, and for each* $a \in \{s_i, t_i\}$*.*

From this new definition of blocking constraints, Friggstad and Gao give the following LP relaxation for UFP-Tree which they prove has an $O(\log(n) \min\{\log(n), \log(k)\})$ integrality gap, which we will call (UFP-TreeRankLP):

$$\begin{aligned}
\max \quad & \sum_{i=1}^{k} w_i x_i && \text{(UFP-TreeRankLP)} \\
\text{s.t.} \quad & \sum_{R_i \in \mathcal{R}, e \in P_i} d_i x_i \leq c_e, && \forall e \in E, \\
& \sum_{R_i \in C(j,v,a)} x_i \leq 1, && \forall \text{ blocking sets } C(j, v, a), \\
& x_i \in [0, 1], && \forall R_i \in \mathcal{R}.
\end{aligned}$$

Additionally, Friggstad and Gao show that the addition of all rank constraints to the natural LP relaxation for UFP-Tree instances results in an $\Omega\left(\sqrt{\log(n)}\right)$ integrality gap through an explicit UFP-Tree instance, similar in spirit to the canonical example in [3]. We provide a brief sketch of this instance.

For each $h \geq 2, h \in \mathbb{N}$, we define a tree T^h by connecting a new root r to the root of a complete 2^{h-1}-arity tree with height $h - 1$. For each $0 \leq i \leq h$, we define L_i to be the set of all vertices on T^h with distance i from r; we say that these vertices are in *level* i. Thus, the number of vertices in T^h is given by

$$n = 1 + \sum_{i=0}^{h-1} \left(2^{h-1}\right)^i = 1 + \frac{\left(2^{h-1}\right)^h - 1}{2^{h-1} - 1} = 1 + \frac{2^{h(h-1)} - 1}{2^{h-1} - 1} \leq 2^{h^2}.$$

This implies that $h \geq \sqrt{\log_2(n)}$. For each edge $e = (u, v)$ with $u \in L_{k-1}$ and $v \in L_k$, let the edge capacity of e be $c_e = 2^{h(h-k+1)}$. Let $\mathcal{R} = \{R_1, \ldots, R_n\}$ be n requests, each R_i of which asks to route $d_i = 2^{h(h-k+1)} - 2^{h(h-k)}$ units of flow from v_i to r, producing $w_i = \frac{1}{2^{(k-1)(h-1)}}$ units of weight. Note that

$$|L_k| = \left(2^{h-1}\right)^{k-1} = 2^{(h-1)(k-1)}.$$

This implies that exactly one unit of weight is evenly distributed among vertices in level k. An example of T^3 is illustrated in Fig. 2.

We call the natural linear programming relaxation (UFP-LP) with all rank constraints (UFP-AllRankLP) and give the following formulation:

$$\max \quad \sum_{i=1}^{k} w_i x_i \qquad \text{(UFP-AllRankLP)}$$

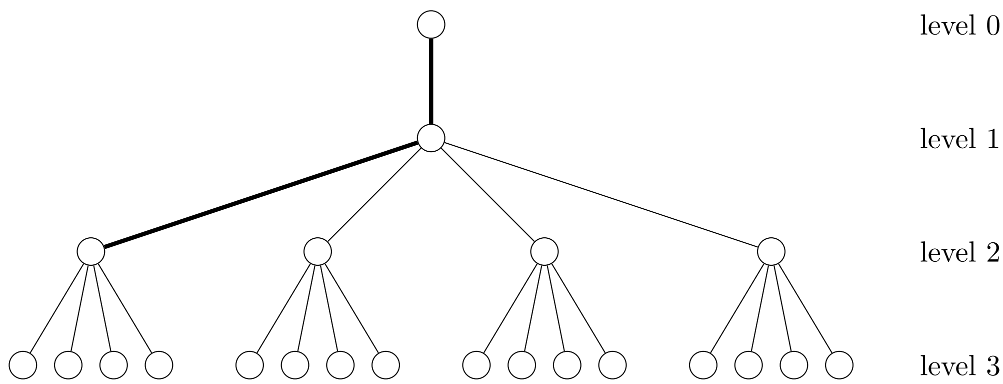

Figure 2: A tree example which shows that the lower bound of the integrality gap of UFP-AllRankTree is $\Omega\left(\sqrt{\log(n)}\right)$. The thick solid line shown in the figure denotes a request from a vertex to the root.

$$\begin{aligned}
\text{s.t.} \quad & \sum_{R_i \in \mathcal{R}, e \in P_i} d_i x_i \le c_e, && \forall e \in E, \\
& \sum_{R_i \in \mathcal{S}} \le \operatorname{rank}(\mathcal{S}), && \forall \text{ intersecting subsets } \mathcal{S} \text{ of requests}, \\
& x_i \in [0,1]. && \forall R_i \in \mathcal{R}.
\end{aligned}$$

Here, we say that a subset $\mathcal{S}$ of requests is *intersecting* if there exists a vertex $v \in V$ on paths P_i for each $R_i \in \mathcal{S}$. To prove that the natural linear programming relaxation (UFP-LP) with all rank constraints has an $\Omega\left(\sqrt{\log(n)}\right)$ lower bound of the integrality gap, Friggstad and Gao first show the integrality lower bound for (UFP-TreeRankLP) using the above example and then invoke the following theorem:

Theorem 5.2. *If x is a feasible solution to* UFP-TreeRankLP *for* UFP-Tree*, then $\frac{x}{9}$ is a feasible solution to* UFP-AllRankLP*.*

As in the canonical path example presented in [3], Friggstad and Gao prove by a simple reverse induction from the leaves to the root that $x_i = \frac{1}{2}, \forall R_i \in \mathcal{R}$ is a feasible fractional solution to (UFP-TreeRankLP) and produces a total weight of $\frac{h}{2}$. Furthermore, they prove that any integral solution to UFP-Tree on T^h produces a total weight of at most 2. To show this, Friggstad and Gao prove the following lemma by another reverse induction from the leaves to the root:

Lemma 5.1. *For each $v \in L_k$, the maximum total weight of a feasible subset of $S(v)$ is at most $\frac{2}{2^{(h-1)(k-1)}}$, where $S(v)$ denotes the set of requests for which s_i is in the subtree rooted at v.*

Note that the optimal fractional solution to (UFP-TreeRankLP) is at least $\frac{h}{2}$. Thus, the integrality gap of (UFP-TreeRankLP) is lower bounded by $\frac{h}{4} = \frac{\sqrt{\log_2(n)}}{4} = \Omega\left(\sqrt{\log_2(n)}\right)$. Since $x_i = \frac{1}{2}, \forall R_i \in \mathcal{R}$ is feasible to (UFP-TreeRankLP), then by Theorem 5.2, $x_i = \frac{1}{18}, \forall R_i \in \mathcal{R}$ is feasible to (UFP-AllRankLP). This proves that the integrality gap of (UFP-AllRankLP) is lower bounded by $\Omega\left(\sqrt{\log(n)}\right)$.

6 UFP-Path LP Relaxations with $O(1)$ Integrality Gap Using DP Embeddings

Anagnostopoulos et al. [1] provide a new LP relaxation for UFP-Path without NBA that employs dynamic programming embeddings into LPs in order to attain an $O(1)$ integrality gap. Prior to this paper the best known integrality gap for UFP-Path was of the order $O(\log(n))$, proven in [6] which additionally conjectured that UFP-Path may in fact have an LP relaxation with an $O(1)$ integrality gap.

Anagnostopoulos et al. provide two LP formulations, one for the unit-weight UFP-Path instance and one for the general UFP-Path instance. In either case, both relaxations are the first for their problem class to have a proven $O(1)$ integrality gap. In the general case, Anagnostopoulos et al. give an extended formulation of the natural LP relaxation that attains an integrality gap bounded by $7 + \epsilon$. It is interesting to note that this integrality gap matches the best known approximation factor (of a polynomial time approximation algorithm) for this problem Bonsma et al. [2].

Martin et al. [8] give a generic method for formulating dynamic programs as linear programs. Anagnostopoulos et al. independently discovered a slightly different and simpler approach for embedding dynamic programs into linear programs, although Anagnostopoulos et al. make mention of its prior discovery. Consequently, Anagnostopoulos et al. design an LP relaxation that uses dynamic programming embeddings for subcases of a UFP-Path instance for which the natural LP relaxation has a large integrality gap; specifically, these subcases are the subset of requests which are big requests (as defined in Section 4 of this survey). The authors remark that dynamic programming embeddings are an uncommon technique, especially in reducing integrality gaps akin to the magnitude of their results from $\Omega(n)$ to $O(1)$. They conjecture that such techniques may become more popular for both proving theoretical results and being employed in mathematical program solvers.

Acknowledgments

We thank Hu Fu for providing the opportunity to survey this area as the term project of CPSC 531F at the University of British Columbia. We also thank Hu Fu for his comments on a presentation of this survey.

References

[1] A. Anagnostopoulos, F. Grandoni, S. Leonardi, and A. Wiese. Constant integrality gap lp formulations of unsplittable flow on a path. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 25–36. Springer, 2013.

[2] P. Bonsma, J. Schulz, and A. Wiese. A constant-factor approximation algorithm for unsplittable flow on paths. *SIAM journal on computing*, 43(2):767–799, 2014.

[3] A. Chakrabarti, C. Chekuri, A. Gupta, and A. Kumar. Approximation algorithms for the unsplittable flow problem. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 51–66. Springer, 2002.

[4] C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity demand flow in a tree and packing integer programs. *ACM Transactions on Algorithms (TALG)*, 3(3):27, 2007.

[5] C. Chekuri, A. Ene, and N. Korula. Unsplittable flow in paths and trees and column-restricted packing integer programs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 42–55. Springer, 2009.

[6] C. Chekuri, A. Ene, and N. Korula. Unsplittable flow in paths and trees and column-restricted packing integer programs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 42–55. Springer, 2009.

[7] Z. Friggstad and Z. Gao. On linear programming relaxations for unsplittable flow in trees. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[8] R. K. Martin, R. L. Rardin, and B. A. Campbell. Polyhedral characterization of discrete dynamic programming. *Operations research*, 38(1):127–138, 1990.