Unsplittable Flow in Paths and Trees and Column-Restricted Packing Integer Programs*

Chandra Chekuri† Alina Ene‡ Nitish Korula§

December 10, 2011

Abstract

We consider the unsplittable flow problem (UFP) and the closely related column-restricted packing integer programs (CPIPs). In UFP we are given an edge-capacitated graph $G = (V, E)$ and k request pairs $R_1, \ldots, R_k$ where each R_i consists of a source-destination pair (s_i, t_i), a demand d_i and a weight w_i. The goal is to find a maximum weight subset of requests that can be routed unsplittably in G. Most previous work on UFP has focused on the *no-bottleneck* case in which the maximum demand of the requests is at most the smallest edge capacity. Inspired by the recent work of Bansal *et al.* [3], we consider UFP on paths as well as trees without the no-bottleneck assumption. We give a simple $O(\log n)$ approximation for UFP on trees when all weights are identical; this yields an $O(\log^2 n)$ approximation for the weighted case. These are the first non-trivial approximations for UFP on trees. We develop a new LP relaxation for UFP on paths that has an integrality gap of $O(\log n)$; previously there was no relaxation with $o(n)$ gap. In contrast we show that the integrality gap of the natural LP has an $O(n/t)$ gap even after applying t rounds of the Sherali-Adams lift-and-project scheme. We also consider UFP in general graphs and CPIPs without the no-bottleneck assumption and obtain new and useful results.

1 Introduction

In the Unsplittable Flow Problem (hereafter, UFP), the input is a graph $G(V, E)$ (directed or undirected; in this paper, we chiefly focus on the latter case) with a capacity c_e on each edge $e \in E$, and a set $\mathcal{R} = \{R_1, R_2, \ldots R_k\}$ of *requests*. Each request R_i consists of a pair of vertices (s_i, t_i), a demand d_i, and a weight/profit w_i. To route a request R_i is to send d_i units of flow along a *single* path (hence the name *unsplittable flow*) in G from s_i to t_i. The goal is to find a maximum-profit set of requests that can be simultaneously routed without violating the capacity constraints; that is, the total flow on an edge e should be at most c_e. A special case of UFP when $d_i = 1$ for all i and $c_e = 1$ for all e is the classical maximum edge-disjoint path problem (MEDP). MEDP has been extensively studied, and its approximability in directed graphs is better understood — the best approximation ratio known is is $O(\min\{\sqrt{m}, n^{2/3}\log^{1/3} n\})$ [25, 40], while it is NP-Hard to approximate to within a factor better than $n^{1/2-\varepsilon}$ [23]; here n and m are the number of vertices and edges respectively in the input graph. For undirected graphs there is a large gap between the known upper and lower bounds on the approximation ratio: there is an $O(\sqrt{n})$-approximation [18] while the best known hardness factor is $\Omega(\log^{\frac{1}{2}-\varepsilon} n)$ under the assumption that $NP \not\subseteq ZPTIME(n^{O(\text{polylog}(n))})$ [1]. Thus UFP is

*A preliminary version of this paper appeared in APPROX 2009 [16].

†Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Partially supported by NSF grants CCF 0728782 and CNS-0721899. chekuri@cs.illinois.edu

‡Dept. of Computer Science, University of Illinois, Urbana, IL 61801. ene1@illinois.edu

§Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Partially supported by NSF grant CCF 0728782. nkorula2@illinois.edu

difficult in general graphs even without the packing constraints imposed by varying demand values; one could ask if UFP is harder to approximate than MEDP. Most of the work on UFP has been on two special cases. One is the uniform capacity UFP (UCUFP) in which $c_e = C$ for all e and the other is UFP with the no-bottleneck assumption (UFP-NBA) where one assumes that $\max_i d_i \leq \min_e c_e$. Note that UCUFP is a special case of UFP-NBA. Kolliopoulos and Stein [27] showed, via grouping and scaling techniques, that certain linear programming based approximation algorithms for MEDP can be extended with only an extra constant factor loss to UFP-NBA. This reduction holds even when one considers restricted families of instances, say those induced by planar graphs. See [20] for a precise definition of when the reduction applies. In [9, 37], a different randomized rounding approach was used for UFP-NBA.

In this paper we are primarily interested in UFP instances that *do not* necessarily satisfy the no-bottleneck assumption. UCUFP and UFP-NBA have many applications and are of interest in themselves. However, the general UFP, due to algorithmic difficulties, has received less attention. One can extend some results for MEDP and UFP-NBA to UFP by separately considering requests that are within say a factor of 2 of each other; this geometric grouping incurs an additional factor of $\log d_{\max}/d_{\min}$ in the approximation ratio, which could be as large as a factor of n [23]. Azar and Regev showed that UFP in directed graphs is $\Omega(n^{1-\varepsilon})$-hard unless $P = NP$; note that the hardness for UFP-NBA is $\Omega(n^{1/2-\varepsilon})$ [23]. Chakrabarti *et al.* [14] observed that the natural LP relaxation has $\Omega(n)$ integrality gap even when G is a path. In contrast, the integrality gap for the path is $O(1)$ for UFP-NBA [14, 20]. One could argue that the integrality gap of the natural LP has been the main bottleneck in addressing UFP.

This paper is inspired by the work of Bansal *et al.* [3] who gave an $O(\log n)$ approximation for UFP on a path. Interestingly, this was the first non-trivial approximation for this problem; previously there was quasi-polynomial time approximation scheme [4], provided the capacities and demands are quasi-polynomially bounded in n. We note that UFP even on a single edge is NP-Hard, since it is equivalent to the knapsack problem. UFP on a path has received considerable attention, not only as an interesting special case of UFP, but also as a problem that has direct applications to resource allocation where one can view the path as modeling the availability of a resource over time. See [6, 8, 11, 14, 4, 3] for previous work related to UFP on a path. The algorithm in [3] is combinatorial and bypasses the $\Omega(n)$ lower bound on the integrality gap of the natural LP. An open problem raised in [3] is whether UFP on trees also has a poly-logarithmic approximation. The difficulty of UFP on paths and trees is not because of routing (there is a unique path between any two nodes) but entirely due to the difficulty of choosing the subset to route[1]. We note that this subset selection problem is easy on a path if $d_i = 1$ for all i (the natural LP is integral since the incidence matrix is totally unimodular) while this special case is already NP-Hard (and APX-Hard to approximate) on capacitated trees [22]. A constant factor approximation is known for UFP-NBA on trees [20]. We prove the following theorem, answering positively the question raised in [3].

Theorem 1.1. *There is an $O(\log n)$ approximation for UFP on n-vertex trees when all weights are equal. There is an $O(\log n \cdot \min\{\log n, \log k\})$ approximation for arbitrary non-negative weights.*

We borrow a crucial high-level idea from [3] of decomposing the given instance into one in which the demands all intersect. We, however, deviate from their approach of using dynamic programming for "large" demands which does not (seem to) generalize from paths to trees; our algorithm for trees is significantly simpler than the complex dynamic programming for the path used by [3]. We show that for the unit-weight case, a greedy algorithm is a 2-approximation if all requests go through a common vertex in the tree. This insight into the performance of the greedy algorithm allows us to develop a new linear programming relaxation for paths.

Theorem 1.2. *There is a linear programming relaxation for UFP on the path that has an integrality gap of $O(\log n)$.*

[1]UFP on paths and trees are perhaps better thought of as pure packing problems than as routing problems. However, we use UFP for historical reasons.

The integrality gap of the relaxation may very well be $O(1)$; resolving this is an interesting open problem. We give some structural results towards resolving this. We underscore the novelty of our relaxation by showing that some reasonable approaches to strengthening the natural relaxation fail to improve the gap. In particular we show that the relaxation obtained after applying t rounds of the Sherali-Adams lift-and-project scheme [36] to the natural relaxation has a gap of $\Omega(n/t)$.

Column-Restricted Packing Integer Programs: UFP on paths and trees are special cases of column-restricted packing integer programming problems (CPIPs). A packing integer program (PIP) is an optimization problem of the form $\max\{wx \mid Ax \leq b, x \in \{0,1\}^n\}$ where A is a non-negative matrix; we use (A, w, b) to define a PIP. A CPIP has the additional restriction that all the non-zero entries in each column of A are identical. It is easy to write UFP on a tree as a CPIP (see Section 5 for formal details). The common coefficient of each column is the "demand" of that column. UFP on general graphs can also be related to CPIPs by using the path formulation and additional constraints [27]. A 0-1 PIP is one in which all entries of A are in $\{0, 1\}$; note that it is also a CPIP. 0-1 PIPs capture the maximum independent set problem (MIS) as a special case and the strong inapproximability results for MIS [24, **?**] imply that no $n^{1-\varepsilon}$-approximation is possible for 0-1 PIPs unless $P = NP$; here n is the number of columns of A. However, an interesting question is the following. Suppose a 0-1 PIP has a small integrality gap because A has some structural properties. For example, if A is totally unimodular, then the integrality gap is 1. What can we say about a CPIP that is derived from A? In other words, one is asking how the "demand version" of a CPIP is related to its "unit-demand" version; this specific question has been asked and addressed previously [27, 35, 20]. A CPIP satisfies the no-bottleneck assumption (NBA) if $\max_j A_{ij} \leq \min_i b_i$. Kolliopoulos and Stein [27] (see also [20] for more general statements) showed that for CPIPs that satisfy the NBA, one can relate the integrality gap of a CPIP to the gap of its underlying 0-1 PIP; there is only an extra constant factor. These ideas are what allows one to relate UFP-NBA to MEDP.

As with UFP, we are interested in this paper in CPIPs where we do not make the NBA assumption. As above, one could ask whether the integrality gap for the demand version of CPIP can be related to its unit-demand version. (Here, we refer to the "natural" relaxation in which one simply relaxes the integrality constraints.) However, the gap example for UFP on the path shows that unlike the no-bottleneck case, such a relationship is not possible. The unit-demand version of UFP on the path has integrality gap 1 while the demand-version has a gap of $\Omega(n)$. It is therefore natural to look for an intermediate case. In particular, suppose we have a CPIP (A, w, b) such that $\max_{i,j} A_{ij} \leq (1 - \delta)b_i$ for each i; this corresponds to the assumption that each demand is at most $(1 - \delta)$ times the *bottleneck* capacity for that demand. We call such a CPIP δ-bounded CPIP. We informally state below a result that we obtain; the formal statement can be found in Section 5.

The integrality gap of a δ-bounded CPIP is at most $O(\log(1/\delta)/\delta^3)$ times the integrality gap of its unit-demand version.

The proof of the above is not difficult and is based on the grouping and scaling ideas of [27]. However, this has not been observed or stated before. We obtain the following result as a consequence.

Corollary 1.3. *For each fixed $\delta > 0$, there is an $O(\log(1/\delta)/\delta^3)$ approximation for UFP on paths and trees if the demand of each request is at most $(1 - \delta)$ times the capacity of the edges on the unique path of the request.*

One class of CPIPs that have been studied before are those in which the maximum number of non-zero entries in any column is at most L. Baveja and Srinivasan [9] showed that the integrality gap of such CPIPs is $O(L)$ if A satisfies the no-bottleneck assumption. In recent and independent work, Pritchard [32] considered PIPs that have at most L non-zero entries per column, calling them L-column-sparse PIPs, and gave an $O(2^L L^2)$ approximation for them. We follow his notation, but obtain a tighter bound by restricting our attention to L-sparse CPIPs.

Theorem 1.4. *There is an $O(L)$-approximation for L-sparse CPIPs via the natural LP relaxation, even without the no-bottleneck assumption. If w is the all 1's vector then a simple greedy algorithm gives an L-approximation to the integral optimum (not necessarily with respect to the LP optimum).*

As corollaries we obtain the following results. We refer to UFP in which the paths for the routed requests have to contain at most L edges as L-bounded-UFP.

Corollary 1.5. *There is an $O(L)$-approximation for L-bounded-UFP in* directed *graphs.*

The demand-matching problem considered by Shepherd and Vetta [35] is an instance of a 2-bounded CPIP and therefore we have.

Corollary 1.6. *There is an $O(1)$-approximation for the demand-matching problem. Moreover, there is a 2-approximation for the cardinality version.*

Note that [35] gives a 3.264 approximation for general graphs and a 3-approximation for the cardinality version, both with respect to the LP optimum. Our $O(L)$ bound for L-bounded CPIPs has a larger constant factor since it does not take the structure of the particular problem into account, however the algorithm is quite simple.

Other Related Work and Discussion: UFP and MEDP are extensively studied and we refer the reader to [1, 18, 19, 21, 25, 26] for various pointers on approximation algorithms and hardness results. Schrijver [34] discusses known results on exact algorithms in great detail. We focus on UFP on paths and trees and have already pointed to the relevant literature. We mention some results on UFP for the special case when $w_i = d_i$. Kolman and Schiedeler [30] considered this special case in directed graphs and obtained an $O(\sqrt{m})$-approximation. Kolman [29] extended the results in [40] for UCUFP to this special case. We note that the $\Omega(n)$ integrality gap for the path [14] does not hold if $w_i = d_i$. In a technical sense, one can reduce a UFP instance with $w_i = d_i$ to an instance in which the ratio $d_{\max}/d_{\min}$ is polynomially bounded. Two approximation techniques for UFP-NBA are greedy algorithms [25, 27, 2] and randomized rounding of the multi-commodity flow based LP relaxation [37, 9, 11, 14]. These methods when dealing with UFP-NBA classify demands as "large" ($d_i \geq d_{\max}/2$) and "small". Large demands can be reduced to uniform demands and handled by MEDP algorithms (since $d_{\max} \leq \min_e c_e$) and small demands behave well for randomized rounding. This classification does not apply for UFP. A simple observation we make is that if we are interested in the cardinality problem then it is natural to consider the greedy algorithm that gives preference to smaller demands; under various conditions this gives a provably good algorithm. Another insight is that the randomized rounding algorithm followed by alteration [38, 14] has good behaviour if we sort the demands in decreasing order of their size — this observation was made in [14] but its implication for general UFP was not noticed. Finally, the modification of the grouping and scaling ideas to handle δ-bounded demands is again simple but has not been noticed before. Moreover, for UFP on paths and trees one obtains constant factor algorithms for any fixed δ. We remark that this result is not possible to derive from the randomized rounding and alteration approach for paths (or trees) because the alteration approach needs to insert requests based on left end point to take advantage of the path structure while one needs the requests to be sorted in decreasing demand value order to handle the fact that we cannot separate small and large demands any more.

Strengthening LP relaxations by adding valid inequalities is a standard methodology in mathematical programming. There are various generic as well as problem specific approaches known. The knapsack problem plays an important role since each linear constraint in a relaxation can be thought of inducing a separate knapsack constraint. Knapsack cover inequalities [12] have been found to be very useful in reducing the integrality gap of covering problems [12, 32]. However, it is only recently that Bienstock [10], answering a question of Van Vyve and Wolsey [39], developed an explicit system of inequalities for the knapsack packing problem (the standard maximization problem) that yields an approximation scheme. Wolsey (as reported in [20]) raises the question of how multiple knapsack constraints implied by the different linear constraints of a relaxation interact since that is what ultimately determines the strength of the relaxation. UFP on a path is perhaps a good test case for examining this question. The $\Omega(n)$ gap example shows the need to consider multiple constraints simultaneously — we hope that our formulation and its analysis is a step forward in tackling other problems.

2 UFP on Trees

Recall that each request R_i consists of a pair of vertices s_i, t_i, a demand d_i and a profit w_i, and if selected, the entire d_i units of demand for this request must be sent along a single path. When the input graph is a tree, there is a *unique* path between each s_i and t_i. For such instances, we refer to this unique path P_i as being the request path for R_i.

The following relaxation is the natural relaxation for UFP on trees: Here, x_i indicates whether the pair $s_i t_i$ is selected.

$$
\begin{array}{rcll}
\max & \sum_{i=1}^{k} w_i x_i & \text{s.t.} & \\
\sum_{i:\, P_i \ni e} d_i x_i & \leq & c_e & (\forall e \in E(G)) \\
x_i & \in & [0,1] & (\forall i \in \{1,\ldots,k\})
\end{array}
$$

This relaxation has an $O(1)$ integrality gap for UFP-NBA on trees [20]. Unfortunately, without NBA, the gap can be as large as $\Omega(n)$ even when the input graph is a path, as shown in [14] (see Section 3). No relaxations with gap $o(n)$ were previously known, even for UFP on paths. The difficulty appeared to lie in dealing with requests for which the demands are very close to the capacity constraints; we confirm this intuition by proving Corollary 1.3 in Section 5: For UFP on trees, if each $d_i \leq (1-\delta)\min_{e \in P_i} c_e$, the natural LP relaxation has an integrality gap of $O(\text{poly}(1/\delta))$. In Section 4, we show how to handle large demands for UFP on paths by giving a new relaxation with an integrality gap of $O(\log n \cdot \min\{\log n, \log k\})$.

In this section, we prove Theorem 1.1, giving a simple combinatorial algorithm to obtain an $O(\log n \cdot \min\{\log n, \log k\})$-approximation for UFP on trees. We first obtain an $O(\log n)$ approximation for unit-profit instances of UFP on trees with n vertices. To do this, we note that if all the request paths must pass through a common vertex, a simple greedy algorithm achieves a 2-approximation.

Lemma 2.1. *Consider unit profit instances of UFP on trees, for which there exists a vertex v such that all request paths pass through v. There exists a 2-approximation algorithm for such instances.*

Proof: Order the requests in increasing order according to their demands. We consider the requests in this order and, if adding the current request maintains feasibility, we add the request to our set.

Let v be a vertex that is on every request path; root the tree T at v. Let OPT be an optimal subset of requests. We will show that OPT routes at most twice as many demands as the Greedy algorithm by using the following exchange argument. Suppose that the smallest request R routed by Greedy has demand d and it uses path P. If this request is also routed by OPT, it follows by induction that OPT routes at most twice as many demands as Greedy. Therefore we may assume that OPT does not route this request. Let x and y be the endpoints of the path P. Let P_1 be the subpath of P from v to x and let P_2 be the subpath of P from v to y. Let e_1 (respectively, e_2) be the last edge (farthest from v) on P_1 (resp. P_2) that has less than d capacity available after routing the requests in OPT. Since the total capacity of e_1 (resp. e_2) is at least d, it follows that there is a request R_1 (resp. R_2) in OPT that uses the edge e_1 (resp. e_2). Since Greedy selected requests in increasing order according to their demand, the requests R_1 and R_2 have demand at least d. Since R_1 and R_2 need to be routed on paths that must pass through v, there will be at least d capacity available on each edge of P_1 and P_2 after routing only the requests in OPT$\setminus\{R_1, R_2\}$. Consequently, $(\text{OPT}\setminus\{R_1, R_2\}) \cup \{R\}$ is feasible. By induction, OPT$\setminus\{R_1, R_2\}$ contains at most twice as many requests as $A\setminus\{R\}$, where A is the set of requests routed by Greedy. □

Lemma 2.2. *There exists an $O(\log n)$-approximation algorithm for unit profit instances of UFP on trees.*

Proof: It is well known that any tree T has a vertex v such that each component of $T\setminus v$ has at most $n/2$ vertices, where n is the number of vertices in T. Moreover, we can find such a vertex (called a *center*) in polynomial time.

Consider a unit profit instance of UFP on a tree T, with request set $\mathcal{R}$. We have the following algorithm for selecting a feasible subset of requests. Let v be a center for T. Let $\mathcal{R}_v \subseteq \mathcal{R}$ be the subset consisting of all requests whose paths pass through the vertex v. Using the 2-approximation algorithm guaranteed by Lemma 2.1, we construct a feasible subset $A \subseteq \mathcal{R}_v$ of these requests. Now let $T_1, ..., T_h$ be the connected components of $T \backslash v$. For each component T_i, we recursively find a feasible set of requests B_i for the UFP instance consisting of the tree T_i and the set of all requests in $\mathcal{R}$ whose paths are completely contained in T_i. Let $B = \bigcup_i B_i$ (note that B is a feasible set since the trees $T_1, ..., T_h$ are disjoint). Finally, we select the largest of the two sets A and B.

Let OPT be an optimal solution; we show that the number of requests selected by the algorithm is at least $\frac{\text{OPT}}{2\log n}$ using induction on the size of the tree. If at least $\frac{|\text{OPT}|}{\log n}$ requests pass through v, by Lemma 2.1 the number of requests in the set A is at least $\frac{|\text{OPT}|}{2\log n}$. Otherwise, the optimal solution has at least $\left(1 - \frac{1}{\log n}\right)|\text{OPT}|$ requests that are completely contained in the connected components of $T \backslash v$. By induction, the number of requests in the set B is at least $\frac{1}{2\log(n/2)}\left(1 - \frac{1}{\log n}\right)|\text{OPT}| = \frac{|\text{OPT}|}{2\log n}$. □

Theorem 1.1 now follows from Lemma 2.2 and Lemma 2.3 below, which is proved using standard profit-scaling.

Lemma 2.3. *Suppose there exists an r-approximation algorithm for unit profit instances of UFP on a given graph. Then there exists an $O(r\min\{\log n, \log k\})$-approximation algorithm for arbitrary instances of UFP on the graph, where k is the number of requests.*

Proof: We group the requests so that the profits of the requests in group i lie in the interval $[2^i, 2^{i+1})$. Clearly, there are at most $2\log k$ groups. For each group, we make the profits of all the requests in the group equal to one and we construct a subset A_i of the requests using the r-approximation algorithm for unit profit instances. Finally, we take the most profitable (according to the initial profits) of the subsets A_i. Let A be this subset. Clearly, the optimal subset for one of the groups has total profit at least $\frac{1}{2\log k}\left(1 - \frac{1}{k}\right)P^*$ (since there are at most $2\log k$ groups and the requests that we removed have profit at most P^*/k). Note that we lose a factor of two by making the profits in each group equal to one and we lose a factor of r by using the r-approximation algorithm for unit profit instances of UFP. Therefore the total profit of the requests in A is at least $\frac{1}{4r\log k}\left(1 - \frac{1}{k}\right)P^* \geq \frac{1}{8r\log k}P^*$. □

3 LP Relaxations for UFP on Paths

The following Linear Programming relaxation is natural for UFP on paths. There is a variable x_i for each request R_i to indicate whether it is selected, and the constraints enforce that the total demand of selected requests on each edge is at most its capacity.

Standard LP $\max \sum_i w_i x_i$

$$\begin{array}{rcll} \sum_{i:\, e\in P_i} d_i x_i & \leq & c_e & (\forall e \in E(G)) \\ x_i & \in & [0,1] & (\forall i \in \{1,\ldots,k\}) \end{array}$$

It is shown in [14] that the integrality gap of this LP relaxation is $\Theta(\log \frac{d_{\max}}{d_{\min}})$ where $d_{\max}$ and $d_{\min}$ are $\max_i d_i$ and $\min_i d_i$ respectively. Unfortunately, this gap can be as bad as $\Omega(n)$, as shown in the following example from [14]; the input path has n edges with edge i having capacity 2^i; request R_i is for 2^i units of capacity on edges i through n, and has profit 1. (See Fig. 1.) An integral solution can only route a single request, for a profit of 1; however, setting $x_i = 1/2$ for each i is a feasible fractional solution to the LP, for a total profit of $n/2$. We refer to this instance as the *canonical integrality gap example*.

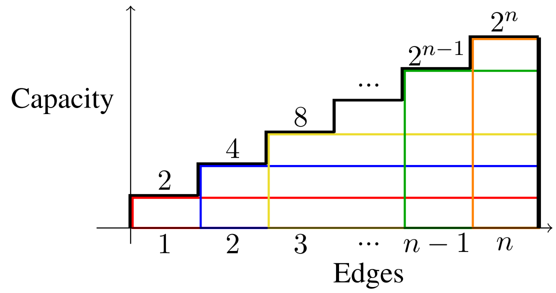

Figure 1: An instance of UFP on paths with large integrality gap.

Though an $O(\log n)$-approximation algorithm for UFP on paths was given in [3], no LP with an integrality gap of $o(n)$ was known for this problem, and obtaining such an LP has been an interesting open question. One could attempt to write a configuration LP for the problem, or to consider strengthening the natural LP, for instance, via the Sherali-Adams hierarchy of relaxations. We show in the rest of this section that these relaxations also have feasible fractional solutions of profit $\Omega(n)$ for the canonical integrality gap example; this can be skipped on first reading without a loss of continuity. For both of the relaxations below, we use $\mathcal{R}_e$ to denote the set of requests passing through edge e.

A Configuration LP: In the configuration LP below, there is a variable $x_{S,e}$ for each set $S \subseteq \mathcal{R}_e$ if the total demand d_S of the requests in S is at most the capacity c_e. Though this LP has an exponential number of variables, we can separate over its dual, which has a polynomial number of variables and constraints that are essentially equivalent to the knapsack problem (with polynomially bounded profits, since we assume that the profits of the original instance are integers in $\{1, \dots, k^2\}$). However, the integrality gap of the configuration LP is also $n/2$, as shown by the canonical example; set $x_i = 1/2$ for each i, and for the jth edge e_j, set $x_{\{R_j\},e_j} = 1/2$, and $x_{S_j,e_j} = 1/2$, where $S_j = \{1, \dots, j-1\}$. (On edge e_1, set $x_{\emptyset,e_1} = 1/2$.)

Config LP $\quad \max \sum_i w_i x_i$

$$
\begin{array}{rcll}
\sum_{S:\, S \subseteq \mathcal{R}_e} x_{S,e} & = & 1 & (\forall e \in E(G)) \\
x_i & \leq & \sum_{S:\, S \subseteq \mathcal{R}_e} x_{S,e} & (\forall i \in \{1, \dots, k\}, e \in P_i) \\
x_{S,e} & \geq & 0 & (\forall e \in E(G), S \subseteq \mathcal{R}_e, d_S \leq c_e)
\end{array}
$$

The Sherali-Adams hierarchy for the Standard LP: For a zero-one programming problem, let P denote the feasible integer polytope, and P_0 denote the convex polytope of an LP relaxation for P. The Sherali-Adams Hierarchy [36] is a sequence $P_0, P_1, P_2 \dots P_n = P$ of (successively tighter) relaxations of P; here $P_i \subseteq P_{i-1}$, and P_n (where n is the number of variables in P) is identical to the original integer polytope. There has recently been significant interest [31, 15] in such hierarchies of linear and semi-definite programming relaxations.

We do not discuss the Sherali-Adams hierarchy in detail here; we simply note that to construct the tth polytope P_t, we first construct a "lifted" polytope P_t^{lift}. In addition to the x_i variables, there is a variable y_S for every set S such that $1 < |S| \leq t$. The constraints of the new polytope P_t^{lift} are formed as follows: For each constraint C of the original polytope, and for each $I, J \subset \{1, \dots, k\}$ such that $I \cap J = \emptyset$ and $|I \cup J| \leq t$, we multiply both the LHS and RHS of constraint C by the polynomial $\prod_{i \in I} x_i \prod_{j \in J}(1 - x_j)$. This produces constraints that are non-linear; we linearize them by replacing each x_i^2 by x_i, and for each monomial of the form $\prod_{i \in S} x_i$ (which must have degree $\leq t$), we replace the monomial by the variable y_S. The polytope P_t is formed by projecting this new polytope down to n dimensions (only retaining the variables x_i); that is, a point $\overline{x}$ is in P_t iff there is a setting of the y_S variables such that $(\overline{x}, \overline{y})$ is in the lifted polytope P_t^{lift}. We refer the reader to [36, 31] for a more complete description of the Sherali-Adams Hierarchy; here, we simply show that the integrality gap of P_t is $\Omega(n/t)$ by giving a feasible solution of profit $n/2t$.

Theorem 3.1. *After applying t rounds of the Sherali-Adams hierarchy to the relaxation **Standard LP**, the integrality gap of the LP obtained is $\Omega(n/t)$.*

Proof: Consider the canonical integrality gap example. For any $t \geq 2$, we set each $x_i = 1/2t$, and each $y_S = 0$ for any set S with $1 < |S| \leq t$. The fractional profit of this solution is $n/2t$, though the integral optimum is 1; it remains only to show that this is a feasible solution for the polytope P_t^{lift}. Constraints of this polytope are formed by picking disjoint sets I, J with $|I \cup J| \leq t$, and multiplying $\prod_{i \in I} x_i \prod_{j \in J}(1 - x_j)$ with constraints of the original polytope. Fix sets I, J, and a constraint C: $\sum_{R_h \in \mathcal{R}_e} d_h x_h \leq c_e$ of the original polytope corresponding to edge e; we argue that the new constraint C' formed by multiplying the LHS and RHS of C by $\prod_{i \in I} x_i \prod_{j \in J}(1 - x_j)$ is satisfied.

First note that if $I > 1$, all variables in the new constraint C' are of the type y_S for some S; as we set each $y_S = 0$, C' is trivially satisfied. If $|I| = 1$, let $I = \{i\}$; removing terms that are equal to 0, we obtain $d_i x_i \leq c_e x_i$ if $R_i \in \mathcal{R}_e$, and $0 \leq c_e x_i$ if $R_i \notin \mathcal{R}_e$. In either case, the constraint C' is obviously satisfied.

It remains only to consider the case $I = \emptyset$; in this case, we are multiplying the original constraint C by $\prod_{j \in J}(1 - x_j)$. After removing all quadratic and higher-order terms, we are multiplying by $(1 - \sum_{j \in J} x_j)$. The LHS of the new constraint C' (ignoring quadratic terms) is $\sum_{R_h \in \mathcal{R}_e, h \notin J} d_h x_h$, and the RHS is $(1 - \sum_{j \in J} x_j)c_e \geq c_e/2$ (as each $x_j = 1/2t$ and $|J| \leq t$). But $\sum_{R_h \in \mathcal{R}_e} d_h x_h \leq \sum_{R_h \in \mathcal{R}_e} d_h/4 < c_e/2$ (where the last inequality is due to the sizes of demands in the canonical integrality gap example), and so the constraint C' is satisfied. □

The two preceding examples show that it is difficult to write an LP relaxation with small integrality gap by only considering "local" constraints, which bound the capacity used on each edge in isolation. A stronger LP needs to introduce constraints that are more global in nature, taking into account that different edges may prevent different subsets of requests from being routed. In Section 4, we give a new LP relaxation for the UFP on paths, and prove that it has a poly-logarithmic integrality gap.

4 New LP Relaxations

We now describe two new LP relaxations for UFP on paths that have an $O(\log n)$ integrality gap. The first relaxation has an exponential number of constraints and is tighter than the second on all instances. However, we do not have an exact separation oracle for this relaxation. The second relaxation is a compact relaxation, and moreover we are able to show that any feasible solution to this relaxation is also feasible for the first when scaled by a constant factor. In fact, our compact relaxation was derived from the first after noticing that it was implicit in an approximate separation oracle for the first.

Corollary 1.3 implies that **Standard LP** has small integrality gap if the demand of each request is small compared to the capacity constraints; recall that in the canonical example with integrality gap $n/2$, every request, if routed, uses the *entire* capacity of the leftmost edge on its path. In the new LP relaxations, we keep the previous constraints to handle "small" requests, and introduce new rank constraints to deal with "big" requests. For each $R_i \in \mathcal{R}$, let the *bottleneck* for R_i be the edge in P_i with least capacity. (If multiple edges have the same minimum capacity, let the bottleneck be the leftmost edge.) Let $\mathcal{S} \subseteq \mathcal{R}$ be the set of all requests R such that the demand of R is smaller than $(3/4) \cdot c(e)$ where e is the bottleneck edge for R. Let $\mathcal{B} = \mathcal{R} \backslash \mathcal{S}$ denote the remaining ("big") requests, and let $\mathcal{B}_e$ denote the set of requests R in $\mathcal{B}$ such that the path for R passes through edge e. For each request R_i, we have a variable x_i denoting whether this request is selected or not. For each set $B \subseteq \mathcal{B}$ of big requests, let $f(B)$ denote the maximum *number* of requests in B that can be simultaneously routed without violating the capacity constraints. For each set B of "big" requests that pass through a common edge, we introduce a *rank* constraint which requires that the total extent to which requests in B are selected by the LP must be at most the number of requests in B that can be routed integrally.

UFP-LP $\max \sum_i w_i x_i$

$$
\begin{array}{rcll}
\sum_{i:\, e \in P_i} d_i x_i & \leq & c_e & (\forall e \in E(G)) \quad \text{[capacity constraints]} \\
\sum_{R_i \in B} x_i & \leq & f(B) & (\forall e \in E(G), B \subseteq \mathcal{B}_e) \quad \text{[rank constraints]} \\
x_i & \in & [0,1] & (\forall i \in \{1, \ldots, k\})
\end{array}
$$

The new constraints are essentially tailor made to mimic the analysis of the greedy algorithm in Section 2 and it is relatively straight forward to show that the integrality gap is $O(\log^2 n)$ [16]. However, it is not obvious that the LP can be solved in polynomial time, even approximately. In [16], we described an approximate separation oracle that allowed one to solve **UFP-LP** to within a constant factor in polynomial time. Subsequently, we noticed that the separation oracle implicitly defined a compact relaxation which includes a polynomial sized subset of the rank constraints. We are able to show an improved bound of $O(\log n)$ on the integrality gap of the compact relaxation, and hence also for **UFP-LP**. We now describe the compact relaxation.

Recall that $\mathcal{B}_e$ is the set of all requests R in $\mathcal{B}$ such that the path for R passes through edge e. Let $\mathcal{B}_{left}(e)$ (resp. $\mathcal{B}_{right}(e)$) be the set of all requests R_i in $\mathcal{B}(e)$ such that the bottleneck edge of R_i is to the left (resp. to the right) of e; if the bottleneck for $R_i \in \mathcal{B}(e)$ is edge e, R_i is in both $\mathcal{B}_{left}(e)$ and $\mathcal{B}_{right}(e)$. For any two requests R_i and R_j in $\mathcal{B}_e$, R_j *blocks* R_i if $d_j > d_i$ and R_i and R_j are not simultaneously routable. For any edge e and any request R_i in $\mathcal{B}_{left}(e)$, let $\mathrm{LeftBlock}(e,i)$ be the set consisting of R_i and all requests R_j in $\mathcal{B}_{left}(e)$ that block R_i. Similarly, for any edge e and any request R_i in $\mathcal{B}_{right}(e)$, let $\mathrm{RightBlock}(e,i)$ be the set consisting of R_i and all requests R_j in $\mathcal{B}_{right}(e)$ that block R_i.

As shown in Corollary 4.11, no two requests in $\mathrm{LeftBlock}(e,i)$ (resp. $\mathrm{RightBlock}(e,i)$) are simultaneously routable. Therefore the rank constraints corresponding to the set $\mathrm{LeftBlock}(e,i)$ and $\mathrm{RightBlock}(e,i)$ forces the LP to fractionally pick at most one of the requests in $\mathrm{LeftBlock}(e,i)$ (resp. $\mathrm{RightBlock}(e,i)$). Using only the rank constraints corresponding to the blocking sets $\mathrm{LeftBlock}(e,i)$ and $\mathrm{RightBlock}(e,i)$, we get the following compact LP.

Compact UFP-LP $\max \sum_i w_i x_i$

$$
\begin{array}{rcll}
\sum_{i:\, e \in P_i} d_i x_i & \leq & c_e & (\forall e \in E(G)) \quad \text{[capacity constraints]} \\
\sum_{R_j \in \mathrm{LeftBlock}(e,i)} x_j & \leq & 1 & (\forall e \in E(G), R_i \in \mathcal{B}_{left}(e)) \quad \text{[blocking constraints]} \\
\sum_{R_j \in \mathrm{RightBlock}(e,i)} x_j & \leq & 1 & (\forall e \in E(G), R_i \in \mathcal{B}_{right}(e)) \quad \text{[blocking constraints]} \\
x_i & \in & [0,1] & (\forall i \in \{1, \ldots, k\})
\end{array}
$$

Since **Compact UFP-LP** contains only a subset of the constraints of **UFP-LP**, it is clear that the former is weaker than the latter. We show that it is not much weaker in the following theorem.

Theorem 4.1. *Let x be any feasible solution to **Compact UFP-LP** on a given instance of UFP on a path. Then there is an absolute constant $\alpha \leq 18$ such that x/α is feasible for **UFP-LP** on the same instance.*

The above theorem implies that the integrality gap of the two relaxations are within a constant factor of each other. We prove the above theorem in Section 4.2. First, we show that the integrality gap of **Compact UFP-LP** is at most $O(\log n)$. We believe that the gap is $O(1)$ and give a connection between the integrality gap

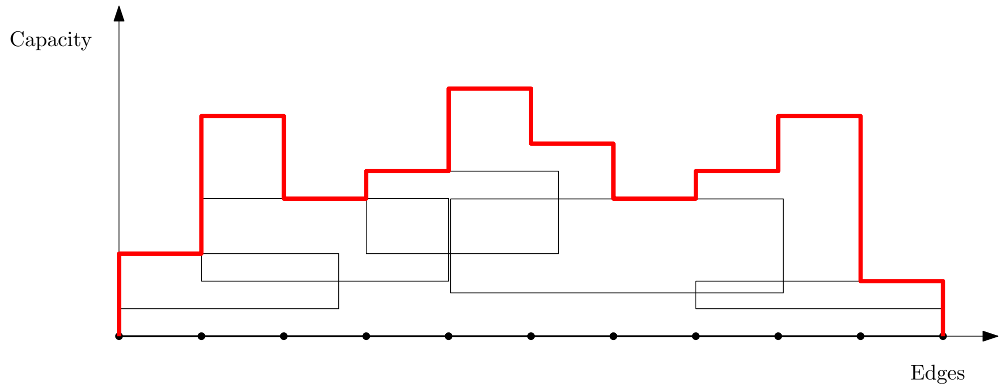

Figure 2: A top-drawn instance of UFP/RIS. The red curve is the capacity profile.

of **UFP-LP** and the integrality gap of the standard LP for the RECTANGLE INDEPENDENT SET (RIS) problem on certain restricted instances that would enable one to prove such an improved bound. Before we define these restricted instances, we give a geometric representation of an UFP instance on paths that was introduced in [7]. Given an UFP instance, we construct a top-drawn drawing of the instance as follows. We represent each request R_i using an axis-parallel rectangle whose start and end points are the start and end points of the request and whose height is equal to the demand d_i. Moreover, we draw the rectangle representing R_i underneath the capacity profile so that its top edge has y-coordinate equal to $c(e)$, where e is the bottleneck edge of R_i (see Figure 4). A set of rectangles that do not overlap when top-drawn gives us a set of requests that is feasible. Bonsma *et al.* [7] show that any feasible UFP solution has a subset of comparable weight whose corresponding top-drawn rectangles do not overlap. We show in the following theorem that there is a similar connection between feasible fractional solutions to **UFP-LP** and feasible fractional solutions to the standard LP for the RIS problem on top-drawn instances. We prove Theorem 4.2 in Section 4.3.

Theorem 4.2. *If the integrality gap of the standard LP for the* RECTANGLE INDEPENDENT SET *problem is at most α for top-drawn instances, the integrality gap of **UFP-LP** is $O(\alpha)$ for instances of UFP on paths in which all requests are big.*

4.1 Bounding the Integrality Gap

In this section, we show an $O(\log n)$ upper bound on the integrality gap of **UFP-LP** and **Compact-LP** for UFP instances on paths.

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

Corollary 4.4. *The LP relaxation **UFP-LP** has $O(\log n)$ integrality gap for instances of UFP on paths.*

Given a fractional solution x to the LP of profit OPT_f, we show how to round it to obtain an integral solution of comparable profit. For any set S of requests, let profit(S), the profit of S be $\sum_{i\in S} w_i x_i$. We round "small" and "big" jobs separately; note that one of profit$(\mathcal{S})$ or profit$(\mathcal{B})$ is at least $\mathrm{OPT}/2$.

Lemma 4.5. *If profit$(\mathcal{S}) \geq \mathrm{OPT}/2$, there is a poly-time algorithm to find an integral solution of value $\Omega(\mathrm{OPT})$.*

Lemma 4.5 follows immediately from Corollary 1.3, as for each request $R_i \in \mathcal{S}$, we have $d_i \leq (3/4)\min_{e\in P_i} c_e$.

The difficulty in bounding the integrality gap of LPs has been in dealing with the "big" requests. However, the blocking constraints allow one to easily prove that there is an integral solution on the big requests with profit comparable to that obtained by these requests in the fractional solution.

We start by establishing an $O(1)$ integrality gap for UFP instances on paths in which all of the request paths end on the last edge. We call such an instance a *one-sided staircase*.

Lemma 4.6. *The LP relaxation* ***Compact UFP-LP*** *has $O(1)$ integrality gap for one-sided staircase instances of UFP on paths.*

Proof: Let x be an optimal fractional solution. We first select a subset of the requests as follows: for each request R_i, we select it independently at random with probability $x_i/4$. Let X be the resulting set. The expected weight of the selected requests is proportional to the weight of the fractional solution but the requests might not form a feasible solution. We pick a feasible subset of the selected requests in two stages: we first pick a subset such that there do not exit two requests such that one blocks the other, and then we select a subset of these requests that is feasible.

We now pick a subset $Y \subseteq X$ such that no two requests such that one request blocks the other as follows. We order the requests from right to left according to their starting point. We add the current request R_i to Y if there does not exist a request $R_j \in Y$ such that one of R_i, R_j blocks the other.

Consider a request R_i in X. We can upper bound the probability that R_i is not in Y as follows. Let C be the set of all requests R_j such that R_j appears before R_i in the ordering and one of R_i, R_j blocks the other. Let C_1 be the subset of C consisting of all requests R_j such that $d_j > d_i$, and let $C_2 = C - C_1$. Note that all requests in C_1 block R_i and therefore $x(C_1) \leq 1$.

Now consider the set C_2. Note that each request in C_2 has demand greater than $3d_i/4$: The starting point of a request R_j in C_2 is at least the starting point of R_i and therefore the bottleneck capacity of R_j is at least $c(e_i)$. Since R_j is big, its demand is greater than $3c(e_i)/4$. Now suppose that there are two requests R_j and R_k in C_2 that are simultaneously routable. Without loss of generality, j starts before k. (Since j and k do not block each other, they cannot start on the same edge.) Consider the first edge e_k of R_k. Since R_k is big for its first edge, there is at most $d_k/3 \leq d_i/3$ capacity available on e_k after routing R_k. But R_j has demand at least $3d_i/4$ and R_j uses edge e_k, which is a contradiction. Therefore, if R is the request in C_2 with minimum demand, all the requests in $C - R$ block R and hence $x(C_2) \leq 1$.

Note that R_i is not in Y only if some request in C was in Y when we considered R_i. The probability that there is some request in C that is in X is $x(C)/4 \leq 1/2$, and therefore $\Pr[R_i \in Y \mid R_i \in X] \geq 1/2$. Thus the expected weight of Y is at least $\sum_i w_i x_i/8$.

Finally, we partition the requests in Y into two feasible subsets as follows. We order the requests in Y from left to right according to their starting point. (Note that since no request blocks another, no two requests start on the same edge.) Let $R_1, R_2, \cdots, R_m$ denote the resulting ordering. We let Y_1 denote the set of all requests R_i in Y such that i is odd, and we let $Y_2 = Y - Y_1$. In the following, we show that Y_1 contains a feasible subset of weight $\Omega(w(Y_1))$; a similar argument shows that Y_2 contains a feasible subset of weight $\Omega(w(Y_2))$.

Let e_i be the start edge of R_i. We claim that the capacities of the odd edges $e_1, e_3, \ldots$ are increasing by a factor of at least three. Since R_i and R_{i+2} are simultaneously routable and they both use edge e_{i+2}, we have $d_i + d_{i+2} \leq c(e_{i+2})$. Therefore $d_i \leq c(e_{i+2}) - d_{i+2} \leq c(e_{i+2})/4$; in the last inequality, we have used the fact that R_{i+2} is big. Since R_i is big, we have $c(e_i) \leq 4d_i/3$ and hence $c(e_i) \leq c(e_{i+2})/3$. Additionally, the demands of the odd requests $R_1, R_3, \ldots$ are increasing by a factor of at least two, since $d_i \geq 3c(e_i)/4 \geq 9c(e_{i-2})/4 \geq 9d_{i-2}/4$. This gives us that, if S is a feasible subset of $\{R_1, R_3, \ldots, R_{i-4}\}$, then $S \cup \{R_i\}$ is also feasible. Let $d(S)$ denote the total demand of the requests in S. We have $d(S) \leq c(e_{i-4}) \leq 2d_{i-4} \leq d_{i-2}$. Since no request blocks another, $d_{i-2} + d_i \leq c(e_i)$, and the claim follows.

We select a subset Z_1 of Y_1 as follows. For each odd index i, we flip a coin; if the coin came up heads, we add R_i to Z_1 if the resulting set remains feasible. It follows from the discussion above that adding R_i to

$Z_1 - \{R_{i-2}\}$ maintains feasibility. Since R_{i-2} is in Z_1 with probability at most $1/2$, R_i is in Z_1 with probability at least $1/4$. Therefore the expected weight of Z_1 is at least $w(Y_1)/4$. This completes the proof of the lemma. □

Corollary 4.7. *The LP relaxation* ***Compact UFP-LP*** *has* $O(1)$ *integrality gap for intersecting instances of UFP on paths.*

Proof: Recall that $\mathcal{B}_{left}(e)$ (resp. $\mathcal{B}_{right}(e)$) is the set of all requests R_i in $\mathcal{B}(e)$ such that the bottleneck edge of R_i is to the left (resp. to the right) of e. (If the bottleneck for $R_i \in \mathcal{B}(e)$ is edge e, R_i can be added to either $\mathcal{B}_{left}(e)$ or $\mathcal{B}_{right}(e)$.) We consider the sets $\mathcal{B}_{left}(e)$ and $\mathcal{B}_{right}(e)$ separately. We map $\mathcal{B}_{left}(e)$ (resp. $\mathcal{B}_{right}(e)$) to a one-sided staircase by making all the requests end (resp. start) at edge e. Finally, we construct an integral solution for each of the one-sided staircase instances using randomized rounding with alteration and we output the most profitable of the two solutions. By Proposition 4.9, the resulting solution is a feasible integral solution for the initial intersecting instance. □

The proof of the following proposition is straightforward and we omit it.

Proposition 4.8. *Let* $\mathcal{B}(\ell)$ *be the set of big requests with path lengths in the interval* $[2^\ell, 2^{\ell+1})$. *There exists an* $\ell \in \{0, 1, \ldots, \log n\}$ *such that profit*$(\mathcal{B}(\ell)) \geq$ *profit*$(\mathcal{B})/\log n$.

Now we can construct an integral solution on the big requests.

Proof of Theorem 4.3: Fix ℓ such that profit$(\mathcal{B}(\ell)) \geq$ profit$(\mathcal{B})/\log n$. Let $L = 2 \cdot 2^{\ell+1}$. Now, pick a random integer p in the range $[1, L]$, and let e_j $(1 \leq j \leq n)$ denote the jth edge on the input path. For each $j \in \lceil n/L \rceil$, let B'_j denote the set of requests in $\mathcal{B}(\ell)$ whose demand paths contain edge e_{p+jL}.

First, note that no request in B'_j overlaps with any request from B'_h if $h \neq j$, as all requests have length at most $L/2$. Further, the probability that any request $R \in \mathcal{B}(\ell)$ is in B'_j for some j is at least $1/4$ (the request R has length at least $L/4$, and we pick edges at an interval L, with a random offset). Therefore, $\mathbb{E}\left[\sum_j \text{profit}(B'_j)\right] \geq \text{profit}(\mathcal{B})/(4\log n)$.

Since each B'_j is an intersecting set, it follows from Corollary 4.7 that B'_j has an integral subset B_j^{int} that can be feasibly routed, of total profit $\Omega(\text{profit}(B'_j))$. Finally, we take the union over all j of B_j^{int}; as requests from B'_j do not overlap those from B'_h for $j \neq h$, we have a feasible integral solution of total profit $\Omega(\text{profit}(\mathcal{B})/(\log n))$. □

4.2 Proof of Theorem 4.1

Let x be any feasible solution to **Compact UFP-LP**. In the following we show that, if there exists a set $B \subseteq \mathcal{B}_e$ such that $x(B) > 18 f(B)$, then x is not a feasible solution for **Compact UFP-LP**.

We first introduce some notation. We define $x(S) = \sum_{R_i \in S} x_i$. Let *left*$(e)$ denote the set of edges to the left of e, together with edge e, and let *right*(e) be the set of edges to the right of e (again including e). Given an edge e and a set $S' \subseteq \mathcal{B}_e$, we say that S' is *feasible on the left* (respectively, on the right), if all requests in S' can be routed simultaneously without exceeding the capacity of any edge in *left*(e) (respectively, *right*(e)). For any set $S \subseteq \mathcal{B}_e$, let $f_\ell(S)$ denote the maximum size subset of S that is feasible on the left and $f_r(S)$ denote the maximum size subset of S that is feasible on the right. (Equivalently, $f_\ell(S)$ is $f(S)$ in the instance obtained by truncating all requests at the right endpoint of e.) Obviously, $f(S) \leq \min\{f_\ell(S), f_r(S)\}$, as any feasible set is feasible on both the left and the right. However, we also note the following simple facts:

Proposition 4.9. *If a set* $S' \subseteq \mathcal{B}_{left}(e)$ *is feasible on the left, it is also feasible on the right.*

Proof: Suppose that not all requests in S' can be simultaneously routed on the right. Order the requests in S' from left to right according to their first (leftmost) edge. Let S'' be a maximum prefix of S' that is feasible on the right and let R_i be the first request in S' that is not in S''. Let d'' denote the total demand of requests in S''. Since $S'' \cup \{R_i\}$ is not feasible on the right, there exists an edge $e^* \in \textit{right}(e) \cap P_i$ such that the total capacity of e^* is less than $d_i + d''$. However, the bottleneck edge e_b for request R_i is in *left*(e), and since all requests in S'' begin at least as far left as R_i and $S'' \cup \{R_i\}$ is feasible on the left, the capacity of e_b is at least $d'' + d_i$. But e_b is the edge of *least* capacity on P_i, and so we obtain a contradiction. □

Corollary 4.10. *For any set $S \subseteq \mathcal{B}_{\textit{left}}(e)$, $f(S) \geq f_\ell(S)$. For any set $S \subseteq \mathcal{B}_{\textit{right}}(e)$, $f(S) \geq f_r(S)$.*

Proof: By symmetry, it suffices to prove the first statement. Let S' be a maximal subset of S that is feasible on the left; that is, $f_\ell(S') = |S'| = f_\ell(S)$. From the previous proposition, all requests in S' can also be simultaneously routed without exceeding the capacity of any edge in $right(s)$. Hence, $f(S') = f_\ell(S)$. But $S' \subseteq S$, and so $f(S) \geq f_\ell(S)$, proving the proposition. □

Corollary 4.11. *For any edge e and any request $R_i \in \mathcal{B}_{left}(e)$, $f(\text{LeftBlock}(e, i)) = 1$. For any edge e and any request $R_i \in \mathcal{B}_{right}(e)$, $f(\text{RightBlock}(e, i)) = 1$.*

Proof: By symmetry, it suffices to show that $f(\text{LeftBlock}(e, i)) = 1$. Suppose for contradiction that there are two requests R_j and R_k in $\text{LeftBlock}(e, i)$ that are simultaneously routable; note that $i \notin \{j, k\}$. Without loss of generality, the starting point of R_j is at most the starting point of R_k. Suppose that R_i and R_k are not simultaneously routable on the left and let e' be an edge in $left(e) \cup \{e\}$ such that $c(e') < d_i + d_k$. Since $d_j > d_i$, we have $c(e') < d_j + d_k$. But R_j also uses e', which contradicts the fact that R_j and R_k are simultaneously routable. Therefore $f_\ell(\{R_i, R_k\}) = 2$. By Corollary 4.10, R_i and R_k are simultaneously routable, which is a contradiction. □

Lemma 4.12. *If there exists a set $B \subseteq \mathcal{B}_e$ such that $x(B) > \alpha f(B)$, there exists a set $B' \subseteq \mathcal{B}_{\textit{left}}(e)$ such that $x(B') > \frac{\alpha}{2} f_\ell(B')$ or a set $B'' \subseteq \mathcal{B}_{\textit{right}}(e)$ such that $x(B'') > \frac{\alpha}{2} f_r(B'')$.*

Proof: Let B_L denote $B \cap \mathcal{B}_{\text{left}}(e)$ and B_R denote $B \cap \mathcal{B}_{\text{right}}(e)$. One of $x(B_L)$ and $x(B_R)$ is at least $x(B)/2 > \frac{\alpha}{2} f(B)$; suppose it is the former. If $f_\ell(B_L) \leq f(B)$, we have $x(B_L) > \frac{\alpha}{2} f_\ell(B_L)$, and we are done. Otherwise, $f_\ell(B_L) > f(B)$. But, from Corollary 4.10, $f(B_L)$ is at least $f_\ell(B_L)$, and so $f(B_L) > f(B)$. But B_L is a subset of B, and hence $f(B) \geq f(B_L)$, which is a contradiction.

Similarly, if $x(B_R) \geq x(B)/2$, the set B_R is such that $x(B_R) > \frac{\alpha}{2} f_r(B_R)$. □

By symmetry, we assume w.l.o.g. that there exists $B' \subseteq \mathcal{B}_{\text{left}}(e)$ such that $x(B') > (\alpha/2) f_\ell(B')$. To complete the proof of Theorem 4.1, we show that if there is a constraint of **UFP-LP** that is violated on the left by a large factor, there is a constraint of **Compact UFP-LP** that is also violated.

Lemma 4.13. *If there exists a set $B \subseteq \mathcal{B}_{\textit{left}}(e)$ such that $x(B) > \beta f_\ell(B)$ for some $\beta > 9$, there exists a set $S' \subseteq B$ such that $f_\ell(S') = 1$ and $x(S') > 1$.*

Proof: Suppose there exists a set $B \subseteq \mathcal{B}_{\text{left}}(e)$ such that $x(B) > \beta f_\ell(B)$. Let $\mathcal{D}_j \subseteq B$ be the set of all requests R_i in B such that d_i lies in the interval $[2^j, 2^{j+1})$. Finally, let B_k be the union of all sets $\mathcal{D}_j$ such that $j \equiv k \pmod 3$. Suppose w.l.o.g. that $x(B_0) \geq x(B)/3$. As $B_0 \subseteq B$, we have $f_\ell(B_0) \leq f_\ell(B)$. Now, for each $j \not\equiv 0 \pmod 3$, set $\mathcal{D}_j = \emptyset$.

Let $\gamma = \max_{S \subseteq B_0} \lfloor x(S)/f_\ell(S) \rfloor$; note that $\gamma \geq 3$. Let $S \subseteq B_0$ be a smallest set that is violated by a factor of γ. It follows from the definition of S that $x(S) \geq \gamma f_\ell(S)$. If $f_\ell(S) = 1$, S is the desired set. Therefore we may assume that $f_\ell(S) > 1$. Let $\mathcal{D}_{\leq j} = \bigcup_{h \leq j} \mathcal{D}_j$. Let j be the smallest index that satisfies $x(\mathcal{D}_{\leq j}) \geq \gamma$.

Sub-Claim. $f_\ell(\mathcal{D}_j) = 1$.

Proof: Suppose two requests R_1 and R_2 in $\mathcal{D}_j$ are simultaneously routable. Suppose without loss of generality that the bottleneck edge e' of R_2 is at least as far right as the bottleneck of R_1. Since $d_2 \geq (3/4)c_{e'}$, the residual capacity of e' after routing R_2 is at most $(1/4)c_{e'} \leq 2^{j+1}/3$. But $d_1 \geq 2^j$, and hence $d_1 + d_2 > c_{e'}$. □

If $x(\mathcal{D}_j) > 1$, $\mathcal{D}_j$ is the desired set. Therefore we may assume that $x(\mathcal{D}_j) \leq 1$, and so $x(\mathcal{D}_{\leq j-3}) \geq \gamma - 1$. If $f_\ell(\mathcal{D}_{\leq j-3}) = 1$, $\mathcal{D}_{\leq j-3}$ is the desired set. Therefore we may assume that $f_\ell(\mathcal{D}_{\leq j-3}) \geq 2$; let R_1 and R_2 be two requests in $\mathcal{D}_{\leq j-3}$ that are simultaneously routable.

Let $S' = S \backslash \mathcal{D}_{\leq j-3}$. Since S' is a proper subset of S, we have $x(S') \leq \gamma f_\ell(S')$ by the minimality of S. Therefore:

$$\gamma f_\ell(S') \geq x(S') = x(S) - x(\mathcal{D}_{\leq j-3}) > \gamma f_\ell(S) - \gamma = \gamma(f_\ell(S) - 1)$$

Since $f_\ell(S') \leq f_\ell(S)$, we must have $f_\ell(S') = f_\ell(S)$. Let S'' be a feasible subset of S' of size $f_\ell(S)$. Let R_3 be a request in S'' whose first edge is farthest left.

Sub-Claim. *The set $S'' \setminus R_3 \cup \{R_1, R_2\}$ is feasible on the left.*

Proof: As $R_3 \in \mathcal{D}_h$ for some $h \geq j$, we have $d_3 \geq 2^j > d_1 + d_2$. On each edge to the left of R_3, R_1 and R_2 can be simultaneously routed, as they do not overlap with any requests in S''. For each edge in P_3, requests R_1 and R_2 together use less capacity than R_3, and so we maintain feasibility. ☐

However, as $|S''| = f_\ell(S)$, $|S'' \setminus R \cup \{R_1, R_2\}| = f_\ell(S) + 1$, which gives a contradiction. ☐

Proposition 4.14. *If there exists a set $S \subseteq \mathcal{B}_e$ such that $x(S) > 1$ and $f(S) = 1$, there exists an i such that $x(Block(e, i)) > 1$. That is, x is not a feasible solution for **Compact UFP-LP**.*

Proof: Let R_i be the request in S with the smallest demand. Since $f(S) = 1$, no two requests in S are simultaneously routable. Therefore $S \subseteq \text{Block}(e, i)$, and hence $x(\text{Block}(e, i)) \geq x(S) > 1$. ☐

4.3 Proof of Theorem 4.2

Let $\mathcal{R}$ be an instance of UFP on paths such that each request $R_i \in \mathcal{R}$ is big. Let $\widehat{\mathcal{R}}$ be the set of top-drawn rectangles representing $\mathcal{R}$. For each request R_i, we let $\widehat{R}_i$ denote the top-drawn rectangle representing R_i.

Lemma 4.15. *Let p be a point and let $\widehat{\mathcal{R}}_p$ the set of all rectangles in $\widehat{\mathcal{R}}$ that contain p. Let $\mathcal{R}_p$ be the set of all requests represented by the rectangles in $\widehat{\mathcal{R}}_p$. We have $f(\mathcal{R}_p) \leq 4$.*

Proof: Let e be the edge of the capacity profile such that p is underneath e. We define a staircase capacity profile c' as follows: for each edge $e' \in left(e)$, we set $c'(e') = \min\{c(e'), c(e'')\}$, where e'' is the edge immediately to the right of e'; for each edge $e' \in right(e)$, we set $c'(e') = \min\{c(e'), c(e'')\}$, where e'' is the edge immediately to the left of e'. Note that, since all of the requests in $\mathcal{R}_p$ use the edge e, a subset $S \subseteq \mathcal{R}_p$ is feasible when the capacities are given by c iff $\mathcal{R}_p$ is feasible when the capacities are given by c'. Moreover, the bottleneck capacity of a request in $\mathcal{R}_p$ is the same for both capacity profiles.

Let $S \subseteq \mathcal{R}_p$ be a feasible set of size $f(\mathcal{R}_p)$. Let $S_\ell \cup \{e\}$ (resp. S_r) be the subset of S consisting of all requests in S whose bottleneck edge is in $left(e)$ (resp. $right(e) \cup \{e\}$). In the following, we show that $f(S_\ell) \leq 2$; a similar argument gives us that $f(S_r) \leq 2$. By Corollary 4.10, $f_\ell(S_\ell) = f(S_\ell)$ and therefore it suffices to show that $f_\ell(S_\ell) \leq 2$.

We modify the requests in S_ℓ by making them end at e. Suppose for contradiction that there are three requests R_1, R_2, R_3 in S_ℓ that are simultaneously routable. Without loss of generality, the starting point of R_i is at most the starting point of R_j for all $i < j$, i.e., the requests appear in the order R_1, R_2, R_3 from left to right. Let e_i be the starting edge of R_i.

As we already noted, the requests R_1, R_2, R_3 are also routable when the capacities are given by c'. Since c' is a staircase capacity profile and the requests in S_ℓ end at e, we can represent the requests R_i using rectangles as follows. The request R_i is represented by a rectangle $\tilde{R}_i$ such that the start and end points of the rectangle are given by the start and end points of the request, and the height is equal to the demand of the request. The y-coordinate of the top edge of the rectangle can be any y-coordinate subject to the constraint that the rectangle fits underneath the capacity profile. We choose the y-coordinates of the rectangles $\tilde{R}_1$, $\tilde{R}_2$, $\tilde{R}_3$ such that they do not overlap; this is possible since the relevant capacity profile is a one-sided staircase, and the requests end at e and they are feasible.

Note that e_i is a bottleneck edge (with respect to c') for R_i. Moreover, in a top-drawn drawing of the rectangles $\tilde{R}_i$, they all contain the point p. Therefore the y-coordinate of p is at most $c(e_1)$. In the current drawing, at least two of the rectangles $\tilde{R}_1$, $\tilde{R}_2$, $\tilde{R}_3$ are below p, i.e., the y-coordinate of their top edge is at most the y-coordinate of p. Let $\tilde{R}_i$ and $\tilde{R}_j$ be two such rectangles. It follows that $d_i + d_j$ is at most $c(e_1)$. Since R_i and R_j are big, we have $d_i > c(e_i)/2 \geq c(e_1)/2$ and $d_j > c(e_j)/2 \geq c(e_1)/2$. Therefore $d_i + d_j > c(e_1)$, which is a contradiction. □

Corollary 4.16. *Let x be a feasible fractional solution to **UFP-LP**$(\mathcal{R})$. Then $x/4$ is a feasible fractional solution to **RIS-LP**$(\widehat{\mathcal{R}})$, where **RIS-LP** is the standard LP for the rectangle independent set problem.*

Since an independent subset of $\widehat{\mathcal{R}}$ is a feasible UFP solution, the theorem follows immediately from the corollary above.

5 UFP and Column-Restricted Packing Integer Programs

In this section, we consider a class of packing problems, so-called *Column-Restricted* Packing Integer Programs (hereafter, CPIP), introduced by Kolliopoulos and Stein [27]. Let A be an arbitrary $m \times n$ $\{0,1\}$ matrix, and d be an n-element vector with d_j denoting the jth entry in d. Let $A[d]$ denote the matrix obtained by multiplying every entry of column j in A by d_j. A CPIP is a problem of the form $\max wx$, subject to $A[d]x \leq b, x \in \{0,1\}^n$, for some integer vectors w, d, b.[2] (Intuitively, a CPIP is a 0-1 packing program in which all non-zero coefficients of a variable x_j are the same.) It is easy to see that the natural LP for UFP in paths and trees is a CPIP: If the path $\mathcal{P}_j$ for request R_j passes through edge e, the coefficient of x_j is demand d_j in the corresponding constraint, and 0 otherwise[3].

CPIPs were studied in [27, 20], and it was shown that the integrality gap of a CPIP with $\max_j d_j \leq \min_i b_i$ is at most a constant factor more than the integrality gap of the corresponding "unit-demand" version; we explain this more formally below, using the notation introduced by [20].

Let P be a convex body in $[0,1]^n$ and $w \in \mathbf{R}^n$ be an objective vector; for any choice of P, w, we obtain a maximization problem $\max\{wx : x \in P\}$. Let γ denote the fractional optimum value of this program, and γ^* denote the optimum *integral* value, which is given by $\max wx$ over all integer vectors $x \in P$. The *integrality gap* of P is γ/γ^*, the ratio between the value of the optimal fractional and integral solutions. A class $\mathcal{P}$ of integer programs is given by problems induced by pairs P, w as above; the integrality gap for a class of problems $\mathcal{P}$ is the supremum of integrality gaps for each problem in $\mathcal{P}$.

We say that a collection of vectors $W \subseteq \mathbf{Z}^n$ is *closed* if for each $w \in W$, replacing any entry w_i with 0 gives a vector $w' \in W$. Subsequently, for each $m \times n$ matrix A and closed collection of vectors W in Z^n, we use $\mathcal{P}(A, W)$ to denote the class of problems of the form $\max\{wx : Ax \leq b, x \in [0,1]^n\}$, where $w \in W$ and b is a vector in $\mathbf{Z}^m_+$. We let $\mathcal{P}^{dem}(A, W)$ denote the class of problems of the form $\max\{wx : A[d]x \leq b, x \in [0,1]^n\}$ where $w \in W, b \in \mathbf{Z}^m_+, d \in \mathbf{Z}^n_+$. Finally, we use $\mathcal{P}^{dem}_{nba}(A, W)$ to denote the class of problems of the same form that satisfy $\max_j d_j \leq \min_i b_i$. For UFP, the condition $\max_j d_j \leq \min_i b_i$ corresponds to the no-nottleneck assumption.

Using techniques introduced in [27], the following theorem was proved in [20]:

[2]If vectors w, d, b are rational, we can scale them as necessary.

[3]In this section we are using the index j for the requests instead of the index i since it is more natural to use i for the rows of A and j for the columns of A.

Theorem 5.1 ([20]). *Let A be a $\{0,1\}$ matrix and W be a closed collection of vectors. If the integrality gap for the collection of problems $\mathcal{P}(A,W)$ is at most Γ, then the integrality gap for the collection of problems $\mathcal{P}^{dem}_{nba}(A,W)$ is at most* $11.542\Gamma \le 12\Gamma$.

The above theorem is used in [20] to give an $O(1)$-approximation for UFP-NBA on trees. Unfortunately, the analogous theorem is not true for $\mathcal{P}^{dem}(A,W)$, as shown by the canonical integrality gap example for the UFP linear program **Standard LP**. In this section, we show that if there exists $\delta < 1$ such that for each i, we have $\max_j A_{ij}d_j \le (1-\delta)b_i$, we *can* obtain an analogous theorem, with integrality gap depending on δ. More precisely, let $\mathcal{P}^{dem}_{\delta}(A,W)$ denote the class of problems of the form $\max\{wx : A[d]x \le b, x \in [0,1]^n\}$ where $w \in W, b \in \mathbf{Z}^m_+, d \in \mathbf{Z}^n_+$, and $\forall i$, $\max_j A_{ij}d_j \le (1-\delta)b_i$. We obtain the following theorem:

Theorem 5.2. *Let A be a $\{0,1\}$ matrix and W be a closed collection of vectors. If the integrality gap for $\mathcal{P}(A,W)$ is at most Γ, the integrality gap for $\mathcal{P}^{dem}_{\delta}(A,W)$ is at most $O(\frac{\log(1/\delta)}{\delta^3}\cdot\Gamma)$.*

Proof: Fix A, w, d, b; given a fractional solution x such that $A[d]x \le b\}$, we show how to obtain an integral solution of total weight comparable to wx. Let $S^t = \{j : d_j \in [(1+\delta)^t, (1+\delta)^{t+1})\}$, and let *dem*$(t) = (1+\delta)^{t+1}$.

Discretize/round up demands and scale down the solution x as follows: For each $j \in S^t$, let $d'_j =$ *dem*(t), and let $x' = \frac{\delta^2}{2(1+\delta)}x$. It follows that $A[d']x' \le \frac{\delta^2}{2}b$, and $wx' = \frac{\delta^2}{2(1+\delta)}wx$.

We now solve a separate "unit-demand" problem for each S^t: For each constraint i, define the load ℓ^t_i on constraint i due to S^t as *dem*$(t)\sum_{j\in S^t} a_{ij}x'_j$. Set b^t_i to be the smallest multiple of *dem*(t) at least as large as ℓ^t_i. Finally, let $d^t_j =$ *dem*(t) if $j \in S^t$, and 0 otherwise; let $w^t_j = w_j$ if $j \in S_t$ and 0 otherwise. We note that $A[d^t]x' \le b^t$, and hence that x' is a feasible fractional solution for the problem $\max\{w^t x : A[d^t]x \le b^t\}$. As each b^t is an integral multiple of *dem*(t) and W is closed, this new problem is in $\mathcal{P}(A,W)$; hence, we can obtain an integral solution z^t of value at least $\frac{w^t x'}{\Gamma}$.

If we could combine the integral solutions z^t for all t, we would have a feasible integral solution of profit wx'/Γ, completing the proof. The capacity used by z^t on constraint i is at most b^t_i, and we have $\sum_t \ell^t_i \le \frac{\delta^2}{2}b_i$. Unfortunately, as b^t_i could be greater than ℓ^t_i (by as much as *dem*(t)), we cannot necessarily combine all z^ts; the unused capacity of $(1-\delta^2/2)b_i$ is not quite enough to account for $\sum_t$ *dem*(t).

Instead, we pick an integer h and let $T(g) = \{t : t \equiv g \pmod h\}$. For some $g \in \{1,\ldots,h\}$, $\bigcup_{t\in T(g)} z^t$ obtains at least $\frac{1}{h}\frac{wx'}{\Gamma}$ of the optimal fractional profit, and by choosing h large enough, we can make sure that the total extra capacity used by each z^t such that $t \in T(g)$ is at most the amount of "unused" capacity.

More precisely, fix a constraint i. Let $t_{\max}(i)$ be the largest t such that $(1+\delta)^t \le (1-\delta)b_i$. (Intuitively, this is the largest t such that demands from S^t can participate in constraint i.) We need to prove that $\sum_{t\in T(g), t\le t_{\max}(i)}$ *dem*$(t) \le (1-\delta^2/2)b_i$. Summing the geometric series (with common ratio $r = (1+\delta)^h$) and using the fact that *dem*$(t_{\max}(i)) \le (1-\delta)b_i \cdot (1+\delta)$ (where the last factor of $(1+\delta)$ comes from rounding up the demands), we obtain:

$$\sum_{t\in T(g), t\le t_{\max}(i)} \mathit{dem}(t) \le \frac{(\mathit{dem}(t_{\max}(i))\cdot r)}{r-1}$$

By picking h large enough ($\frac{2}{\delta}\ln(2/\delta^2)$), we can ensure that this sum is at most $(1-\delta^2/2)b_i$. By picking the best shift $g \in \{1,\ldots,h\}$, we obtain an integral solution of value $\frac{1}{h}\frac{wx'}{\Gamma} = \frac{\delta}{2\ln(2/\delta^2)}\frac{\delta^2}{2\Gamma}wx$. Therefore, the integrality gap of this LP is $\Omega(\frac{\ln(1/\delta)}{\delta^3}\cdot\Gamma)$. □

Thus, we obtain Corollary 1.3 as a special case of Theorem 5.2.

5.1 Bounded-Length Paths and Sparse Columns

Recall that an L-column-sparse CPIP is one in which there are at most L non-zero entries in each column; UFP with bounded-length paths can be viewed as a column-sparse CPIP. Several papers have considered EDP and UFP when the path lengths are restricted to some given length L; this restriction is meaningful in many applications. In addition, one can show that in expanders and related graphs, any fractional solution can be converted into another solution with "short" (polylogarithmic in n) flow paths. We refer to this restricted version of UFP as L-bounded-UFP. There is an $O(L)$-approximation and integrality gap for L-bounded UFP with NBA [37, 27, 14]. Surprisingly, this natural restriction does not seem to have been explored for UFP without NBA. Here we observe that an $O(L)$-approximation can be obtained for this more general setting as well.

First, we consider the unit-profit case and a simple greedy algorithm that sorts the pairs in non-decreasing order of demand values and routes the jth pair if there is sufficient residual capacity for it on a path of length at most L.

Lemma 5.3. *The greedy algorithm is an L-approximation for unit-profit instances of L-bounded-UFP.*

Using profit-scaling we obtain the following.

Corollary 5.4. *There is an $O(L \log k)$-approximation for L-bounded-UFP.*

One can, however, do better, proving Corollary 1.5 to get an $O(L)$-approximation with arbitrary profits by noting that L-bounded UFP is an example of an $(L+1)$-column-sparse CPIP: We can write a multi-commodity flow based LP relaxation for L-bounded-UFP where we insist that the flow for each pair (s_j, t_j) is only along paths in $\mathcal{P}_j^L$ where an (s_j, t_j) path π is in $\mathcal{P}_j^L$ iff π is of length at most L and all edges on π have capacity at least d_j. The LP is given below:

$$\begin{array}{rcll} \max & \sum_{j=1}^k w_j \sum_{\pi \in \mathcal{P}_j^L} f_\pi & & \text{s.t.} \\ \sum_{\pi \in \mathcal{P}_j^L} f_\pi & \leq & 1 & j = 1, \ldots, k \\ \sum_{j=1}^k \sum_{\pi \in \mathcal{P}_j^L : \pi \ni e} d_j f_\pi & \leq & c_e & e \in E(G) \end{array}$$

This LP can be solved using the ellipsoid method since the separation oracle for the dual is the bounded length shortest path problem. Further, it is column-restricted and $(L+1)$-column-sparse.[4] Thus, Corollary 1.5, giving an $O(L)$-approximation for L-bounded UFP in *directed* graphs, follows from Theorem 1.4.

It remains only to prove Theorem 1.4, obtaining an $O(L)$-approximation for L-column-sparse CPIPs. The proof of this theorem is essentially implicit in previous work [14].

Proof of Theorem 1.4: We briefly describe the algorithm of [14]. Given an optimal fractional solution x, we perform randomized rounding, with alteration as follows: For the jth variable, we set it to 1 with probability $x_j/4k$; we say that such a variable is *selected*. We now pick a subset of the selected variables to be *accepted*, and output this set. We prove that a variable is accepted with probability at least $1/2$, provided that the variable is selected. Thus, the expected value of the final integral solution is at least $1/8k$ times the optimal fractional value.

To determine which variables are accepted, order the selected variables in *non-increasing* order of their demand values d_j. Process the variables in this order and accept a variable if adding it to the set of variables

[4] To make it column-restricted, we should technically write constraints as $\sum_{\pi \in \mathcal{P}_j^L} d_j f_\pi \leq d_j$, for each j.

already accepted does not violate any constraint. In order to show that any selected variable is accepted with probability at least $1/2$, it suffices to prove that the probability that any constraint it participates in prevents it from being accepted is at most $1/2k$.

Consider a constraint i, and suppose $d_j \leq b_i/2$. Since the expected capacity on i used by selected variables is at most $b_i/4k$, the probability that the selected variables use more than $b_i/2$ capacity is less than $1/2k$ by Markov's inequality. Thus, the probability that there is insufficient capacity for j is at most $1/2k$. If $d_j > b_i/2$, consider the variables with demand at least d_j, as only they can prevent j from being accepted. It is easy to see that the probability that any of these variables is selected is at most $1/2k$; if none of them is selected, there is sufficient capacity for j. □

6 Conclusions

We conclude with some open problems:

- What is the integrality gap of the new relaxation **UFP-LP** for paths? We show it is $O(\log n \min\{\log n, \log k\})$, but know of only a constant-factor lower bound. Is there a matching (or even logarithmic) upper bound? A *staircase* instance for UFP is one such as the canonical integrality gap example, in which all request paths go through a common edge. (Hence w.l.o.g., capacities increase monotonically, and then decrease.) Can one obtain an improved integrality gap for UFP on staircases?

- Can the approximation ratio for UFP on trees be improved from $O(\log n \min\{\log n, \log k\})$ to $O(\log n)$ or $O(\log k)$, matching the $O(\log n)$ approximation of [3] for UFP on paths? Theorem 5.2 / Corollary 1.3 imply that the difficulty is in dealing with requests that occupy almost all the capacity of their bottleneck edge.

- The (exponential-sized) LP relaxation **UFP-LP** can be extended to UFP on trees, and it has small integrality gap. However, finding a separation oracle appears to be difficult. Can the techniques of Section 4.2 be extended to this case?

References

[1] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. *Proc. of IEEE FOCS*, 226–241, 2005.

[2] Y. Azar, O. Regev. Combinatorial algorithms for the unsplittable flow problem. *Algorithmica*, 441(1):49–66, 2006. Preliminary version in *Proc. of IPCO*, LNCS, 15–29, 2001.

[3] N. Bansal, Z. Friggstad, R. Khandekar, M. R. Salavatipour. A logarithmic approximation for unsplittable flow on line graphs In *Proc. of ACM-SIAM SODA*, 702–709, 2009.

[4] N. Bansal, A. Chakrabarti, A. Epstein, B. Schieber. A Quasi-PTAS for unsplittable flow on line graphs In *Proceedings of the 38th Aannual ACM Symp. on Theory of Computing*, 721–729, 2006.

[5] N. Bansal, N. Korula, V. Nagarajan, A. Srinivasan. On k-Column Sparse Packing Programs. arXiv.org preprint. `arXiv:0908.2256v3`. `http://arxiv.org/abs/0908.2256`.

[6] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, B. Schieber. A unified approach to approximating resource allocation and scheduling. *J. of the ACM* 48(5): 1069–1090, 2001. Preliminary version in *Proc. of ACM STOC*, 735–744, 2000.

[7] P. Bonsma, J. Schulz, A. Wiese. A constant factor approximation algorithm for unsplittable flow on paths. *Proc. of IEEE FOCS*, 2011.

[8] A. Bar-Noy, S. Guha, J. Naor, B. Schieber. Approximating the Throughput of Multiple Machines in Real-Time Scheduling. *SIAM J. on Comput.* 31(2):331–352, 2001. Preliminary version in *Proc. of ACM STOC*, 622–631, 1999.

[9] A. Baveja, A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Math. Oper. Res.*, 25(2):255–280, 2000.

[10] D. Bienstock. Approximate formulations for 0-1 knapsack sets. *Oper. Res. Lett.* 36(3): 317–320, 2008.

[11] G. Calinescu, A. Chakrabarti, H. Karloff, Y. Rabani. Improved approximation algorithms for resource allocation. In *Proceedings of the 9th Integer Programming and Combinatorial Optimization Conference*, volume 2337 of *Lecture Notes in Computer Science*, 439–456, 2001.

[12] R.D. Carr, L. Fleischer, V.J. Leung, C.A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. *Proc. of ACM-SIAM SODA*, 106–115, 2000.

[13] R. Carr and S. Vempala. Randomized meta-rounding. *Random Structures and Algorithms*, 20(3):343–352, 2002.

[14] A. Chakrabarti, C. Chekuri, A. Gupta, A. Kumar. Approximation Algorithms for the Unsplittable Flow Problem *Algorithmica* 47(1):53–78, 2007.

[15] M. Charikar, K. Makarychev, and Y. Makarychev. Integrality Gaps for Sherali-Adams Relaxations. *Proc. of ACM STOC*, 2009.

[16] C. Chekuri, A. Ene, and N. Korula. UFP in Paths and Trees and Column Restricted Packing Integer Programs. *Proc. of APPROX*, 2009.

[17] C. Chekuri, S. Khanna. Edge disjoint paths revisited. *Trans. on Algorithms* 4(3), 2007. Preliminary version in *Proc. of ACM-SIAM SODA*, 2003.

[18] C. Chekuri, S. Khanna, F. B. Shepherd. An $O(\sqrt{n})$ Approximation and Integrality Gap for Disjoint Paths and Unsplittable Flow. *Theory of Computing*, Vol 2, 137–146, 2006.

[19] C. Chekuri, S. Khanna, F. B. Shepherd. Edge-Disjoint Paths in Planar Graphs with Constant Congestion. *Proc. of ACM STOC*, 757–766, 2006.

[20] C. Chekuri, M. Mydlarz, F. B. Shepherd. Multicommodity Demand Flow in a Tree and Packing Integer Programs. *ACM Trans. on Algorithms* 3(3), 2007. Preliminary version in *Proc. of ICALP*, 410–425, 2003.

[21] J. Chuzhoy, V. Guruswami, S. Khanna, K. Talwar. Hardness of Routing with Congestion in Directed Graphs. *Proc of ACM STOC*, 165–178, 2007.

[22] N. Garg. V. V. Vazirani, M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica* 18(1):3–20, 1997.

[23] V. Guruswami, S. Khanna, B. Shepherd, R. Rajaraman, M. Yannakakis. Near-Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems. *J. of Computer and System Sciences*, 67(3): 473–496, 2003. Preliminary version in *Proc. of ACM STOC*, 1999.

[24] J. Håstad. Clique is Hard to Approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[25] J. M. Kleinberg. Approximation algorithms for disjoint paths problems. PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1996.

[26] S.G. Kolliopoulos. Edge-disjoint Paths and Unsplittable Flow. In Handbook of Approximation Algorithms and Metaheuristics, ed. T. F. Gonzalez, Chapman & Hall/CRC, 2007.

[27] S. G. Kolliopoulos, C. Stein. Approximating disjoint-path problems using greedy algorithms and Packing Integer Programs. *Math. Programming A*, (99):63–87, 2004. Preliminary version in *Proc. of IPCO*, 1998.

[28] S.G. Kolliopoulos, N.E. Young. Approximation algorithms for covering/packing integer programs. *J. of Computer and System Sciences*, 71(4): 495–505, 2005.

[29] P. Kolman. A Note on the Greedy Algorithm for the Unsplittable Flow Problem. *Information Processing Letters*, 88(3):101–105, 2003.

[30] P. Kolman., C. Scheideler. Improved bounds for the unsplittable flow problem. *J. of Algorithms* 61(1): 20–44, 2006. Preliminary version in *Proc. of ACM-SIAM SODA*, 2002.

[31] C. Mathieu, A. Sinclair. Sherali-Adams relaxations of the matching polytope. To appear in *ACM STOC*, 2009.

[32] D. Pritchard. Approximability of Sparse Integer Programs. *Proc. of ESA*, 2009.

[33] P. Raghavan. Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. *J. Computer and System Sciences* 37(2): 130–143, 1988.

[34] A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency. Springer-Verlag, 2003.

[35] B. Shepherd, A. Vetta. The Demand Matching Problem. *Math. of Operations Research*, 32(3): 563–578, 2007. Preliminary version in *Proc. of IPCO*, 2002.

[36] H. Sherali, W. P. Adams. A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems. *Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, Vol. 52, 1994.

[37] A. Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. *Proc. of IEEE FOCS*, 416–425, 1997.

[38] A. Srinivasan. New Approaches to Covering and Packing Problems. *Proc. of ACM-SIAM SODA*, 567–576, 2001.

[39] M. Van Vyve, L.A. Wolsey. Approximate Extended Formulations. *Math. Programming* 105: 501–522, 2006.

[40] K. Varadarajan, G. Venkataraman. Graph Decomposition and a Greedy Algorithm for Edge-disjoint Paths. *Proc. of ACM-SIAM SODA*, 379–380, 2004.