Approximation Algorithms for the Unsplittable Flow Problem

Amit Chakrabarti, Chandra Chekuri, Anupam Gupta, and Amit Kumar

Abstract. We present approximation algorithms for the unsplittable flow problem (UFP) in undirected graphs. As is standard in this line of research, we assume that the maximum demand is at most the minimum capacity. We focus on the non-uniform capacity case in which the edge capacities can vary arbitrarily over the graph. Our results are:

- We obtain an $O(\Delta^{-1}\log n)$ approximation ratio for UFP, where $n$ is the number of vertices, $\Delta$ is the maximum degree, and $\alpha$ is the expansion of the graph. Furthermore, if we specialize to the case where all edges have the same capacity, our algorithm gives an $O(\Delta^{-1}\log n)$ approximation.
- For certain strong constant-degree expanders considered by Frieze [17] we obtain an $O(\sqrt{\log n})$ approximation for the uniform capacity case.
- For UFP on the line and the ring, we give the first constant-factor approximation algorithms.

All of the above results improve if the maximum demand is bounded away from the minimum capacity. The above results either improve upon or are incomparable with previously known results for these problems. The main technique used for these results is randomized rounding followed by greedy alteration, and is inspired by the use of this idea in recent work.

Key Words. Unsplittable flow problem, Disjoint paths, Approximation algorithms, Randomized rounding, Alteration, Expanders, Line networks, Ring network.

1. Introduction. In the unsplittable flow problem (UFP), we are given an $n$-vertex graph $G = (V, E)$ with edge capacities $\{c_e\}$, and a set of $k$ vertex pairs (terminals) $T = \{(s_i, t_i): i = 1, \ldots, k\}$; each pair $(s_i, t_i)$ in $T$ has a demand $p_i$ and a weight (or profit) $w_i$. The goal is to find the maximum weight subset of pairs from $T$, along with a path for each chosen pair, so that the entire demand for each such pair can be routed on its path while respecting the capacity constraints.

Let us note at the outset that even very special cases of UFP are NP-hard: for instance, when $G$ is just a single edge, UFP specializes to the KNAPSACK problem. When each
$c_e = 1$ and each $\rho_i = w_i = 1$, UFP specializes to the well-known maximum edge-disjoint paths problem (EDP), the goal being simply to find the largest number of pairs from $T$ which can be simultaneously connected by edge-disjoint paths in $G$. EDP is NP-hard even when restricted to planar graphs.

A substantial amount of research has focused on obtaining good approximation algorithms for both EDP and UFP due to their importance in network routing and design. For EDP, the best known approximation ratio on general graphs is $O(\min(n^{2/3}, \sqrt{m}))$ [14], where $n$ and $m$ are the number of vertices and edges in the graph, respectively. In directed graphs the best known approximation ratio is $O(\min(n^{2/3} \log^{3/3} n, \sqrt{m}))$ [35] and it is NP-hard to approximate it to a ratio better than $\Omega(n^{1/2})$ [20]. However, in undirected graphs, which are the focus of this paper, EDP is only known to be hard to approximate to within constant factors [19]. Very recently, a hardness factor of $\Omega(\log^{1/2 - \epsilon} n)$ has been shown [3], [4]. Improved approximation ratios for EDP have been obtained for special classes of graphs like trees, mesh-like planar graphs, and graphs with high expansion; see, e.g., [21] and [25] for references.

Let $\rho_{\text{max}} = \max_i \rho_i$ be the maximum demand among the pairs and let $c_{\text{min}} = \min_e c_e$ be the minimum capacity of an edge. In this paper we only consider instances with $\rho_{\text{max}} \leq c_{\text{min}}$; this is a standard assumption in the literature and is sometimes referred to as the no-bottleneck assumption. In its absence, a UFP instance on a graph $G = (V, E)$ can be embedded into any other graph $G' = (V, E')$ with $E \subseteq E'$, thus making it difficult to study the role of graph structure in the approximability of the problem. Moreover, the restriction is a reasonable one in many applications: e.g., it still includes EDP as a special case. In the rest of the paper we assume without loss of generality that $c_{\text{min}} = 1$ and that $0 < \rho_i \leq 1$ for all $i$.

A special case of UFP is the uniform capacity unsplittable flow problem (UCUFP) in which all edges have the same capacity. UCUFP has received more attention and its approximability is often related to the corresponding EDP problem; much less is known about UFP where edges have varying capacities.

1.1. Our Results. In this paper we address UFP with non-uniform edge capacities on undirected graphs. Our results are quantified in terms of the so-called flow number $^6 F_G$ of the underlying graph $G$; this parameter was defined by Kolman and Scheideler [25], who related $F_G$ to the expansion of the graph, and showed that $F_G = O(\Delta \alpha^{-1} \log n)$, where $\alpha$ is the edge expansion and $\Delta$ is the maximum degree of the graph $G$. We now present our results for general graphs; while comparisons to known results are given in Section 1.3, we mention that our results improve upon, or are incomparable with, previous results:

- An $O(F_G \log n) = O(\Delta \alpha^{-1} \log^2 n)$ approximation for UFP.
- An $O(F_G) = O(\Delta \alpha^{-1} \log n)$ approximation algorithm for UCUFP.

---

^6 We formally define the flow number in Section 2.2, but we point out that $F_G$, as used in this paper, depends only on the structure of the underlying graph $G$ and not on the edge capacities.

^7 Kolman and Scheideler actually gave a definition for flow number that could take non-uniform edge capacities into account as well. In this paper flow number is for the underlying graph, independent of capacities.
When the maximum demand is much smaller than the smallest capacity, the above bounds can be improved. In particular, if \( \rho_{\text{max}} \leq \epsilon_{\text{min}} / B \) for some integer \( B \), the approximation guarantees improve to \( O((F_G \log n)^{1/B}) = O((\Delta \alpha^{-1} \log^2 n)^{1/B}) \) for UFP, and \( O(F_G^{1/B}) = O((\Delta \alpha^{-1} \log n)^{1/B}) \) for UCUFP.

In fact, we have a continuum of approximation ratios between UFP and UCUFP of the form \( O(F_G \cdot \min(\log n, c_{\text{max}})) \), where \( c_{\text{max}} \) is the maximum capacity of an edge (assuming \( c_{\text{min}} = 1 \)). The above results are typically most interesting when \( G \) is a constant-degree expander, with \( \Delta \alpha^{-1} = O(1) \); however, as noted in [25], there are other interesting cases such as butterflies and hypercubes where \( F_G \) can be shown to be a polylogarithmic factor better than the upper bound implied by \( F_G = O(\Delta \alpha^{-1} \log n) \).

Additionally, we obtain even better approximation ratios on special classes of graphs by further exploiting some of the techniques used in proving the above. In particular, we obtain:

- An \( O(\sqrt{\log n}) \) approximation for UCUFP on “sufficiently strong” constant degree expanders as defined by Frieze [17] (see Definition 2.2 and Theorem 4.1).
- An \( O(1) \) approximation for UFP on line and ring networks (see Theorems 5.5 and 5.8).

1.2. Techniques. Previous approaches to approximating EDP and UCUFP on graphs with high expansion relied on proving the existence of near-optimal solutions to the multicommodity flow relaxation of the problem that use short flow paths (i.e., those that are only polylogarithmic in length). Korman and Scheideler [25] generalize this to UFP through their notion of flow number \( F \). However, their upper bound on the length of the flow paths depends on the edge-capacities in \( G \), which could be quite large in some cases, giving a weaker bound. We take a different approach, and show the existence of flow paths using only a few (polylogarithmic number of) edges of low capacity, even though the overall length of the flow path might be large. Since high capacity edges (of capacity \( \Omega(\log n) \)) behave fairly well under randomized rounding, this leaves us to worry only about the behavior of the low capacity edges under randomized rounding.

Our second idea, which subsequently proves useful for the case of the line and the ring as well, is to perform the randomized rounding step with more care. Naive rounding schemes scale down the fractional solution before randomized rounding, with the scaling factor chosen to be large enough to argue that none of the constraints are violated. Typically, the events corresponding to the violation of these constraints are not independent and the union bound is too weak to estimate the failure probability of the randomized rounding. Hence, probabilistic tools like the Lovász Local Lemma (as in [33] and [25]) or a correlation inequality like the FKG inequality (as in [33]) are used to overcome these problems. Not surprisingly, these approaches are often technically involved, adding substantial complexity to both the algorithm and the analysis. We take a different route, and use the method of alterations [2] which is applicable to monotone problems. Applications of this technique to approximation algorithms were recently given by Srinivasan [34], who applied it to general packing and covering problems, and by Calinescu et al. [13] who applied it to a specific packing problem. In this approach the first step is the same as above: scaling followed by
randomized rounding. However, instead of desiring feasibility (i.e., that all constraints be satisfied by the randomized rounding), one looks at the random solution and alters it if it is not feasible: in our case this is done by changing certain ones in the random solution to zeros to ensure feasibility. Since this greedy (problem-dependent) alteration step ensures feasibility by fiat, the burden shifts to analyzing the expected loss in quality during the alteration step. This turns out to be simple and effective for various problems, and we believe that this idea will find more applications in the future.

1.3. Relationship to Previous Work. In this section we discuss previous work on UFP, UCUFP, and EDP, and also indicate how our results mentioned above relate to previously known results.

Culminating a long line of work, Frieze [17] recently showed that for regular expanders with sufficiently strong expansion and sufficiently large (but constant) degree, there exists a constant $c$ such that any $cn/\log n$ vertex pairs can be connected via edge-disjoint paths provided no vertex appears in more than a constant (depending on, and less than, the degree) number of pairs. This result is optimal to within constant factors, and has also been extended to expander digraphs [11]. An immediate consequence of this is an $O(\log n)$ approximation for EDP on such expanders. In 1996 Kleinberg and Rubinfeld [22] had used an earlier result of Broder et al. [12] to show that a deterministic online algorithm, the so-called bounded greedy algorithm (BGA), gave an $O(\log n \log \log n)$ approximation guarantee for EDP on such expanders. (In fact, Frieze’s result mentioned above implies an $O(\log n)$ bound for BGA.) In the same paper, Kleinberg and Rubinfeld also showed the existence of a near-optimal fractional solution to any multicommodity flow instance on an expander that used only short paths of length $O(\log^3 n)$. This latter result formed the basis of an $O(\log^3 n)$ approximation for UCUFP on expanders by Srinivasan [33]. While the above results do not explicitly specify the dependence of the approximation ratio on $\Delta$ and $\alpha$, Kolman and Scheideler [25] suggest that the actual approximation ratio is $\Omega(\Delta^2 \alpha^{-2} \log^3 n)$.

In the context of UCUFP, the results of Kleinberg and Rubinfeld on short flow paths were improved by Kolman and Scheideler [24], [25]. Their results were stated in terms of a parameter $F_{G,c}$ of a graph $G$, which we call the capacitated flow number of $G$; here $c$ refers to the capacities of the edges. They proved the existence of near-optimal solutions to multicommodity flow instances that use paths of length $O(F_{G,c})$. (The earlier paper [24] gave its results in terms of a parameter called the routing number $R$, but these results were superseded by those given in [25] in terms of $F_{G,c}$.) Moreover, in addition to improving the approximation ratio for UCUFP, the results of Kolman and Scheideler offered other advantages: the dependence on the expansion $\alpha$ and the maximum degree $\Delta$ were improved and made explicit, the bound on the flow path lengths was strengthened, and the proof, based on the work of Leighton and Rao [26], was much simpler and direct.

For our results, we use a quantity $F_G$, which we call the flow number of $G$; in contrast to $F_{G,c}$, this quantity depends only on the structure of the graph $G$ and is independent of the edge capacities. Also, while the two parameters $F_G$ and $F_{G,c}$ have similar definitions, their values turn out to be incomparable in general. (A special case
Approximation Algorithms for the Unsplittable Flow Problem

when $F_G = F_{G,e}$ is when all edges have unit capacities—i.e., when we have an instance of UCUFP.

We now relate our results to the previous known results for UFP and UCUFP:

- Our approximation guarantee for UFP is $O(F_G \log n) = O(\Delta^{-1} \log^2 n)$, which is independent of the edge capacities in the network; this is incomparable with the best known approximation ratio of $O(F_{G,e})$ given in Theorem 4.1 of [25].

- $O(F_G) = O(\Delta^{-1} \log n)$ approximation ratio for UCUFP, which matches the bound of $O(F_{G,e})$ given by Theorem 4.1 of [25]; however, we achieve this approximation via a different algorithm.

- For the case where the maximum demand is a factor $1/B$ smaller than the smallest capacity, our bounds are $O((F_G \log n)^{1/B})$ and $O(F_G^{1/B})$ for UFP and UCUFP, respectively. When compared with the bound of $O(B(F_G^{1/B} - 1))$ given in Theorem 4.5 of [25], our UFP bound is incomparable while the UCUFP bound is better by a factor of $B$.

- The $O(\sqrt{\log n})$ approximation for UCUFP on “sufficiently strong” constant degree expanders is the first sub-logarithmic approximation for constant degree expanders that we are aware of, and improves on the current best approximation ratio of $O(\log n)$ [17], [25].

**UFP on the line:** The EDP problem on the line corresponds to the maximum independent set problem on interval graphs, which has a polynomial time algorithm. However, UCUFP on the line generalizes KNPACK and hence is NP-hard; in fact, it is equivalent to the task assignment problem on a single machine with fixed time windows. Generalizations of the task assignment problem to multiple machines and time windows have been studied in the recent past [7], [29], and most of these problems have $O(1)$ approximation algorithms as well as $O(1)$ integrality gaps.

This is not the case with UFP on the line, for which no constant-factor approximation was known before this work. In fact, if the demands are not constrained to be less than the minimum capacity, the integrality gap of the natural linear programming relaxation for UFP could be $\Omega(\min(\log \rho_{\text{max}}, n))$ (see Theorem 5.7). Furthermore, two of the standard techniques used to develop $O(1)$ approximations for the task assignment problem, i.e., the local-ratio method [8], [7] and rounding fractional solutions [29], seem not to extend to the case of UFP. In this work we build upon ideas of Calinescu et al. [13], and combine dynamic programming and randomized rounding with alterations to give the first constant-factor approximation for UFP on the line when $\rho_{\text{max}} \leq \epsilon_{\text{min}}$. For the general case (without the no-bottleneck assumption), we give an algorithm with approximation ratio $\Omega(\log \rho_{\text{max}})$ which matches the integrality gap in Theorem 5.7. We extend the results on the line to the ring via a simple reduction.

2. Preliminaries

2.1. The Natural LP Relaxation. UFP has a natural integer programming formulation based on multicommodity flow. Let $\mathcal{P}_i$ denote the set of all paths in $G$ from $u_i$ to $v_i$. The
integer program (IP) is

$$\max \sum_{i=1}^{k} w_i x_i, \quad \text{s.t.} \quad \sum_{\pi \in P_i} f_{\pi} = x_i, \quad i = 1, \ldots, k,$$

$$\sum_{i=1}^{k} \sum_{\pi \in P_i : \pi \ni e} \rho_i f_{\pi} \leq c_e, \quad e \in E(G),$$

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, k,$$

$$f_{\pi} \in \{0, 1\}, \quad \pi \in \bigcup_{i=1}^{k} P_i.$$
**DEFINITION 2.2 (Strong Expansion).** A $\Delta$-regular $n$-vertex graph $G$ is said to be an $(\alpha, \beta, \gamma)$-expander, for parameters $\alpha \in (0, 1 - \beta)$, $\beta \in (0, 1)$, and $\gamma \in (0, \frac{1}{2})$, if, for any subset $U \subseteq V(G)$, we have

$$|U| \leq \gamma n \implies |\partial U| \geq (1 - \beta)\Delta |U|$$

and

$$\gamma n < |U| \leq n/2 \implies |\partial U| \geq \alpha \Delta |U|.$$  

We say that $G$ is a **strong expander** if it is an $(\alpha, \beta, \gamma)$-expander for some constants $\alpha, \beta, \gamma$ with $\beta$ sufficiently small.

Basically, this definition implies that the graph, besides having expansion $\alpha$ as in Definition 2.1, has even better expansion $(1 - \beta)$ on “small” sets of vertices. Note that the condition of being an $(\alpha, \beta, \gamma)$-expander gets stronger as $\beta$ decreases. As noted by Frieze [17], random regular graphs and Ramanujan graphs [27] are examples of strong expanders.

Finally, we define the **flow number** $F_G$ of a graph $G = (V, E)$, a concept first used by Kolman and Scheideler [25]. Let $\text{deg}(x)$ denote the degree of vertex $x \in V$. Consider the following concurrent multicommodity flow instance defined on $G$: let all edges in $G$ have unit capacity, and let the demand between any pair of vertices $u, v \in V$ be $\text{deg}(u) \text{deg}(v)/(2|E|)$ for $u \neq v$.

For any solution $S$ to this instance, define the flow value to be the maximum $\lambda$ such that $S$ routes at least a $\lambda$ fraction of every demand in the instance. Let the dilation $D(S)$ be the length of the longest flow path in $S$, and let the congestion $C(S)$ be the inverse of the flow value; in other words, we have to scale the edge capacities by $C(S)$ to ensure that all demands are satisfied by this flow $S$. The flow number $F_G$ is defined as the minimum, over all possible solutions $S$, of the quantity $\max\{C(S), D(S)\}$.

An important result proved in Theorem 2.4 of [25] is the following: if $G$ has edge-expansion $\alpha$, and maximum degree $\Delta$, then

$$\Omega(\alpha^{-1}) \leq F_G \leq O(\Delta \alpha^{-1} \log n).$$

For their results on UFP in [25], Kolman and Scheideler extended the definition of the flow number to handle non-uniform capacities. This was done essentially by replacing each edge by $\lceil c_e \rceil$ parallel copies, and letting the **capacitated flow number** $F_{G,c}$ be the flow number of the resulting graph.\(^8\)

It is worth noting that the quantities $F_{G,c}$ and $F_G$ are incomparable. Indeed, consider the line graph on $n$ nodes, where all edges except the middle edge in the line have capacity $c$, and the middle edge has capacity $1$. It can be verified that $F_G$ is $\Theta(n)$, but $F_{G,c} = \Theta(cn) \gg F_G$. On the other hand, consider the balanced binary tree on $n$ nodes, where all edges at level $i$ have capacity $n/2^i$; in this case $F_G = \Theta(n)$, while $F_{G,c} = \Theta(\log n) \ll F_G$.

---

\(^8\) While paper [25] refers to this quantity merely as $F$, we call it $F_{G,c}$ to emphasize the dependence on the capacities $c_e$ as well as the graph $G$. 
Finally, we often make use of the following Chernoff–Hoeffding bound (see [28]).

**Theorem 2.3 (Chernoff–Hoeffding).** Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [0, 1]$ for all $i$. Consider $X = \sum_{i=1}^n X_i$. Let $\mu$ denote $E[X]$. For any $\delta > 0$,

$$\Pr[X \geq (1 + \delta)\mu] < \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}}\right)^\mu.$$

### 3. Approximation Bounds for UFP Based on Expansion.

As indicated in Section 1.2, our approach is to show that the flows in any fractional solution to LPMAIN can be rerouted to yield a new fractional solution in which the flow paths use few edges of small capacity, where “few” is quantified using the flow number $F_G$; we call such a solution **favorable**. While such a rerouting may reduce the profit of the resulting solution, we prove that this loss can be made small. Next, we show that any favorable fractional solution can be rounded efficiently to an integral solution without much reduction in the profit.

#### 3.1. Rerouting Using Short Paths.

The idea of rerouting to use short flow paths is not new, having been first given by Kleinberg and Rubinfeld [22], and subsequently widely used. Our contribution is to redefine the notion of “short”: we restrict our attention only to edges of low capacity, requiring our paths to have few edges of small capacity. Note that this rerouting procedure need not be algorithmically efficient: we merely want to establish the existence of a favorable fractional solution with high profit. If we know such a solution exists, it can be obtained by solving a modified version of LPMAIN in which we have $f_\pi$ variables defined only for paths which use few edges of small capacity. As mentioned in Section 2.1, such an LP can be solved exactly in polynomial time using the ellipsoid method or approximately solved using efficient combinatorial algorithms.

Before getting into the technical details, let us recall that $F_G = O(\Delta^{-1} \log n)$, and that the capacities and demands have been normalized so that $c_{\min} = 1$ and $\rho_{\max} \leq 1$.

**Definition 3.1 (Favorable Solution).** A fractional solution to LPMAIN is said to be $(c, d)$-favorable if every flow path used by the solution has at most $d$ edges of capacity at most $c$.

**Theorem 3.2.** For parameters $\varepsilon \in (0, 1]$ and $c \geq 1$, given a fractional solution to LPMAIN with profit $W$, there exists a $(c, \frac{4cF_G}{\varepsilon})$-favorable fractional solution with profit at least $W/(1 + \varepsilon)$.

The following corollary will be useful in the context of UCUFP; it is obtained by setting $c = 1$ in Theorem 3.2.

**Corollary 3.3.** For any $\varepsilon \in (0, 1]$, given a fractional solution to LPMAIN with profit $W$, there exists a $(1, \frac{4F_G}{\varepsilon})$-favorable fractional solution with profit at least $W/(1 + \varepsilon)$.
The remainder of this section is devoted to the proof of Theorem 3.2; the reader more interested in the rounding of favorable solutions should jump to Section 3.2.

**Proof of Theorem 3.2.** Our proof uses the concept of a balanced multicommodity flow problem (BMFP) defined by Kolman and Scheideler [25]. For our purposes, a BMFP instance consists of an uncapacitated graph, a set of ordered vertex pairs \( \{(u_i, v_i)\} \), and demands \( 0 \leq \rho_i \leq 1 \), one for each vertex pair. The total demand entering a vertex \( v \) is defined as the sum of \( \rho_i \) for all \( i \) such that \( v_i = v \); the total demand leaving a vertex \( v \) is defined similarly. In a BMFP instance, the total demand entering or leaving a vertex \( x \) is required to be equal to its degree \( \deg(x) \).

Suppose, as in the statement of Theorem 3.2, that we are given a fractional solution to LPMAIN with profit \( W \). Let \( \mathcal{P} \) be the set of all flow paths used by this solution. Set \( L = 2cF_G/\varepsilon \). Let \( \mathcal{P}' \) denote the subset of \( \mathcal{P} \) consisting of paths with at least \( 2L \) edges of capacity at most \( c \). We now define an instance of BMFP on the underlying uncapacitated graph \( G \). For each flow path \( \pi \in \mathcal{P}' \), if \( \pi \in \mathcal{P}_1 \), we “orient” it from \( s_i \) to \( t_i \), and for vertex \( u \) on \( \pi \), let \( \text{pred}_u(u) \) denote the vertex that is the predecessor of \( u \) on \( \pi \). We say that \( u \) is a good vertex if \( \text{pred}_u(u) \) exists and the edge \( (\text{pred}_u(u), u) \) has capacity at most \( c \). Let \( u_1, u_2, \ldots, u_L \) be the first \( L \) good vertices on \( \pi \), and let \( v_1, v_2, \ldots, v_L \) be the last \( L \) good vertices on \( \pi \). We add the pairs \( \{(u_j, v_j); 1 \leq j \leq L\} \), each with demand \( \rho_i f_{\pi_i}/c \), to the BMFP instance. We do this for all the flow paths in \( \mathcal{P}' \). Since each edge \( e \) incident to a vertex \( x \) can contribute at most \( \min(c_e/c, 1) \) to the demand entering or leaving \( x \), the total demand entering or leaving any vertex is clearly at most its degree. We then add dummy demands, if required, to satisfy the definition of a BMFP. We now need the following proposition, which appears in [25, Claim 2.2]:

**Proposition 3.4.** A \( 1/(2F_G) \) fraction of all the demands (i.e., each demand scaled down by \( 1/(2F_G) \)) in a BMFP can be concurrently satisfied on the underlying uncapacitated graph \( G \) using a family of flow paths of length at most \( 2F_G \) each.

Let \( \mathcal{Q} \) be a family of flow paths guaranteed by Proposition 3.4. We take the flow going over paths in \( \mathcal{P}' \) and use these paths in \( \mathcal{Q} \) to reroute this flow. Note that a path \( \pi \in \mathcal{P}' \) is associated with \( L \) paths in \( \mathcal{Q} \), each of which “shortcuts” \( \pi \). We send \( \rho_i f_{\pi_i}/L \) flow through each of these shortcuts, adjusting the flow on edges in \( \pi \) appropriately. When we do this for all paths, we obtain a candidate fractional solution with profit \( W \) that uses paths with at most \( \max(L + 2F_G, 2L) \) edges of capacity at most \( c \). Notice that \( L + 2F_G \leq 2L \), if \( \varepsilon \leq 1 \). Thus, the flow paths in this candidate solution have at most \( 2L = 4cF_G/\varepsilon \) edges with capacity at most \( c \).

This candidate solution could violate some edge capacities. However, by Proposition 3.4, had we sent \( \rho_i f_{\pi_i}/(2cF_G) \) flow through each shortcut for \( \pi \in \mathcal{P}' \), we would have had a total flow of at most \( 1 \) on each edge. Since we are in fact sending \( \varepsilon \rho_i f_{\pi_i}/(2cF_G) \) flow over each shortcut, we get a total flow of at most \( \varepsilon \) on each edge due to the shortcuts. Thus, after adding the flow paths in \( \mathcal{P}' \setminus \mathcal{P} \), the total flow on an edge \( e \) is at most \( c_e + \varepsilon \leq (1 + \varepsilon)c_e \). Now, scaling each flow value and each \( x_i \) by \( 1/(1 + \varepsilon) \) gives us a feasible solution. The new profit after scaling is clearly \( W/(1 + \varepsilon) \), which proves Theorem 3.2. \( \square \)
3.2. Rounding a Favorable Solution

THEOREM 3.5. For a large enough value of $d$, given a $(\log n, d)$-favorable fractional solution to LPMAIN with profit $W$, we can efficiently compute a (random) integral solution with expected profit $W'$ such that:

1. $W' = \Omega(W/d)$.
2. If, additionally, $\rho_{\text{max}} \leq 1/B$, for integer $B \geq 2$, then $W' = \Omega(W/d^{1/(B-1)})$.
3. If, additionally, each $\rho_i = 1/B$, for integer $B \geq 1$, then $W' = \Omega(W/d^{1/B})$.

Before proving this theorem, let us derive some of the results implied by it.

COROLLARY 3.6. For graphs with expansion $\alpha$ and maximum degree $\Delta$, the following can be derived:

(a) An $O(F_G \log n) = O(\Delta \alpha^{-1} \log^2 n)$ approximation for UFP.
(b) An $O(F_G) = O(\Delta \alpha^{-1} \log n)$ approximation for UCUFP.
(c) For integer $B$, if $\rho_{\text{max}} \leq 1/B$, then the approximation ratios improve to $O((F_G \log n)^{1/B}) = O((\Delta \alpha^{-1} \log^2 n)^{1/B})$ for UFP, and $O(F_G^{1/B}) = O((\Delta \alpha^{-1} \log n)^{1/B})$ for UCUFP.

PROOF OF COROLLARY 3.6. For Part (a), applying Theorem 3.2 with $c = \log n$ and $\epsilon = 1$ gives us a $(\log n, O(F_G \log n))$-favorable fractional solution; now applying Part 1 of Theorem 3.5 completes the proof.

For Part (b), note that Corollary 3.3 gives a $(1, O(F_G))$-favorable solution; however, since all edges in UCUFP have unit capacity, such a solution is trivially also $(\log n, O(F_G))$-favorable. Now applying Part 1 of Theorem 3.5 completes the proof.

In fact, the above ideas can be combined to give an $O(\min\{\log n, c_{\text{max}}\} F_G)$ approximation for UFP; we can apply Theorem 3.2 with $c = \min\{\log n, c_{\text{max}}\}$ and $\epsilon = 1$ to get a $(c, O(F_G))$-favorable solution, interpret it as an $O(\log n, O(c F_G))$-favorable solution, and finally apply Part 1 of Theorem 3.5 to get the $O(c F_G)$ approximation.

We turn to proving Part (c). Suppose we have an instance $\mathcal{I}$ of UFP with $\rho_{\text{max}} \leq 1/B$ with integer $B \geq 2$. Let us create two new UFP instances, $\mathcal{I}_1$ and $\mathcal{I}_2$, both with the same underlying graph as $\mathcal{I}$ but with $\mathcal{I}_1$ having exactly those source–sink pairs from $\mathcal{I}$ with demands at most $1/(B+1)$ and $\mathcal{I}_2$ having the remaining source–sink pairs, i.e., those with $\rho_i \in (1/(B+1), 1/B]$.

Part 2 of Theorem 3.5 now gives us an $O(d^{1/B})$ approximation for $\mathcal{I}_1$ from a $(\log n, d)$-favorable fractional solution to the LP relaxation of $\mathcal{I}_1$. To approximate $\mathcal{I}_2$, we create a new instance $\mathcal{I}_2'$ by setting all demand values in $\mathcal{I}_2$ to exactly $1/B$. Since we have scaled each demand by a factor of at most $(B+1)/B$, we can take an optimum solution to the LP relaxation of $\mathcal{I}_2$, divide each $x_i$ by $(B+1)/B$, and end up with a feasible solution to the LP relaxation of $\mathcal{I}_2'$ that has a $B/(B+1)$ fraction of the profit of the former. Using Part 3 of Theorem 3.5, we can then get an integral solution to $\mathcal{I}_2'$ of profit at least $\Omega(d^{-1/B}) \times B/(B+1)$ times the optimum profit of the LP relaxation of $\mathcal{I}_2$. However, this integral solution is also feasible for $\mathcal{I}_2$, since we only increased demands in going from $\mathcal{I}_2$ to $\mathcal{I}_2'$; thus we have an $O(d^{1/B})$ approximation for $\mathcal{I}_2$. 
Either $I_1$ or $I_2$ has optimal profit at least half that of $I$, so we can simply pick the better of the two approximate solutions. Finally, for UFP, we can set $d = O(F_G \log n)$ by applying Theorem 3.2; for UCUFP, Corollary 3.3 implies that we can set $d = O(F_G)$.

**Remark.** The above corollary implies a constant factor approximation for UCUFP when $\rho_{\max} \leq 1/\log F_G$. If $\rho_{\max} \leq 1/\max\{\log F_G, \log \log n\}$, then a constant factor approximation can be obtained for UFP as well. Thus, in cases when $F_G = O(\log n)$, such as when $G$ is a butterfly or an expander, UFP has a constant factor approximation algorithm when $\rho_{\max} = O(1/\log \log n)$. Our algorithms are considerably simpler than those in [25] that achieve similar results. Also, our proofs (as the reader will soon see) are substantially simpler than those in [25] which rely upon the Lovász Local Lemma.

### 3.2.1. Rounding in the general case

In this section and the following one, we prove the several parts of Theorem 3.5. Our rounding is based on the work of Srinivasan [34]: we randomly round the $(\log n, d)$-favorable fractional solution after appropriate scaling, and follow that by an alteration phase to obtain a feasible solution. We prove that this yields an $O(d)$ approximation in expectation. We note that Srinivasan [33] and Baveja and Srinivasan [9] showed that randomized rounding yields an $O(d)$ approximation for UCUFP if all flow path lengths are bounded by $d$. However, the proof is involved and is based on the FKG inequality. While it is conceivable that those techniques can be used to round the favorable solutions guaranteed by the previous section, we believe that such an approach would be more involved than the one we present below.

**Proof of Theorem 3.5 (Part 1).** The rounding procedure works in two phases: the selection phase, where we choose random paths, and the pruning or alteration phase, in which we ensure that our solution is feasible.

**Selection Phase:** Independently, for each $i \in \{1, \ldots, k\}$, we do the following: we select at most one of the paths in $P_i$ with the property that each path $\pi \in P_i$ is selected with probability equal to $f_{\pi} / (16d)$. To do this, we order the paths in $P_i$ arbitrarily as $\pi_1, \pi_2, \ldots, \pi_h$. For $0 \leq j \leq h$ define $y_j$ as $(1/(16d)) \sum_{\ell \leq j} f_{\pi_{\ell}}$. Now pick a random number $\zeta$ from $[0, 1]$. We select path $\pi_j$ iff $\zeta \in [y_{j-1}, y_j)$; if $\zeta \geq y_h$, no path is selected. Since $\sum_{\pi \in P_i} f_\pi = x_i$, we will have selected some path in $P_i$ with probability $x_i / (16d)$.

**Alteration/Pruning Phase:** For path $\pi \in P_i$, let $\rho(\pi)$ denote the demand value $\rho_i$ associated with $\pi$. We sort the paths in descending order of their demand values. We consider the paths picked in the selection phase, one by one in the sorted order above. When considering a path $\pi$ we either add it to the final solution or discard it. The criterion for adding a path $\pi$ to the solution is as follows: if $\pi$ can be added to the current set of paths without violating edge capacities, add it, else discard it. If some path $\pi$ is added, it should be understood that the demand $\rho(\pi)$ is routed along $\pi$.

It is clear that we will have a feasible integral solution at the end of the pruning phase. The following lemma suffices to prove Part 1 of Theorem 3.5.
LEMMA 3.7. The resulting random integral solution has expected profit $\Omega(W/d)$.

Before proving this lemma, we state a technical probabilistic tail estimate whose proof we defer to the Appendix.

LEMMA 3.8. Let $a_1, \ldots, a_h, y_1, \ldots, y_h \in [0, 1]$ be such that $\forall i < j, (a_i \leq \frac{1}{2} \Rightarrow a_j \leq \frac{1}{2})$ and furthermore $\sum_{i=1}^{h} a_i y_i \leq 1$. Let $0 < \theta < 1$ and let independent $0$-$1$ random variables $Y_1, \ldots, Y_h$ and (possibly dependent) $0$-$1$ random variables $Z_1, \ldots, Z_h$ be defined as follows:

$Y_i = \begin{cases} 1, & \text{with probability } \theta y_i, \\ 0, & \text{otherwise,} \end{cases}$

$Z_i = \begin{cases} 1, & \text{if } \sum_{j<i} a_j Y_j \leq 1 - a_i, \\ 0, & \text{otherwise.} \end{cases}$

Then, for each $i$, $\Pr[Z_i = 0]$ is at most:

1. $(2 + 2e)\theta$, in general.
2. $e^{B\theta} B^{-1}$, if each $a_i \in [0, 1/B]$ for an integer $B \geq 2$.
3. $e^{B\theta} B$, if each $a_i = 1/B$ for an integer $B \geq 1$.

PROOF OF LEMMA 3.7. Recall that the pruning phase of the algorithm orders the paths used by the $(\log n, d)$-favorable solution in a specific way, based upon the demand values associated with the paths, thereby effectively ordering the $k$ commodities. Without loss of generality, assume that this ordering is $1, 2, \ldots, k$. Now, for each commodity $i$, edge $e$, and path $\pi \in \bigcup_{i=1}^{k} \mathcal{P}_i$, define indicator random variables for the following events of interest:

- $X_i(\pi)$, for the event “the selection phase chooses path $\pi \in \mathcal{P}_i$”.
- $Y_i(e)$, for the event “the selection phase chooses a path in $\mathcal{P}_i$ that contains $e$”.
- $Z_i(e)$, for the event “even if all commodities preceding $i$ were routed along the paths chosen for them in the selection phase, there would be at least $\rho_i$ residual capacity on edge $e$”.
- $A_i(\pi)$, for the event “the final random solution (after the pruning phase) routes commodity $i$ along path $\pi$”.
- $A_i$, for the event “commodity $i$ is routed in the final random solution”.

Note that the pruning phase eliminates a path $\pi$ chosen in the selection phase only if $Z_i(e) = 0$ for some $e \in \pi$. Therefore,

$\Pr[A_i(\pi) = 1] \geq \Pr[X_i(\pi) = 1 \land \forall e \in \pi, (Z_i(e) = 1)]$

$\geq \Pr[X_i(\pi) = 1] \cdot \left(1 - \sum_{e \in \pi} \Pr[Z_i(e) = 0 \mid X_i(\pi) = 1]\right)$

$= \Pr[X_i(\pi) = 1] \cdot \left(1 - \sum_{e \in \pi} \Pr[Z_i(e) = 0]\right),$

where the final equation follows because $Z_i(e)$ is independent of the routing of commodity $i$, by definition. We already know that $\Pr[X_i(\pi) = 1] = f_\pi/(16d)$, so we turn our attention to upper bounding $\Pr[Z_i(e) = 0]$. 

Fix an edge \( e \in E(G) \). The random variables \( \{Y_i(e)\}_{i=1}^k \) are easily seen to be independent from the description of the rounding algorithm. Also, setting \( \gamma_i(e) = \sum_{\pi \in \mathcal{P}_i : \pi \ni e} f_{\pi} \) and \( \theta = 1/(16d) \), we get \( \Pr[Y_i(e) = 1] = \theta \gamma_i(e) \). Setting \( a_i(e) = \rho_i/c_e \) and using the constraints in LPMAIN, we see that, for all \( i, a_i(e), \gamma_i(e) \in [0, 1] \) and that \( \sum_{i=1}^k a_i(e) \gamma_i(e) \leq 1 \). Translating the definition of \( Z_i(e) \) into algebraic notation directly gives us \( Z_i(e) = 1 \) iff \( \sum_{j<i} a_j(e) Y_j(e) \leq 1 - a_i(e) \). Finally, the ordering of the paths before the pruning phase ensures that, \( \forall i < j, (a_i(e) \leq \frac{1}{2} \Rightarrow a_j(e) \leq \frac{1}{2}) \). Therefore, we may apply Part 1 of Lemma 3.8 to obtain

\[
\Pr[Z_i(e) = 0] \leq \frac{2 + 2e}{16d}.
\]

For those edges that satisfy \( c_e > \log n \), we can say something stronger. Since \( \rho_{\max} \leq 1 \), the random variables \( \{\rho_i Y_i(e)\}_{i=1}^k \) are distributed in \([0, 1]\). Let \( Y = \sum_{j=1}^k \rho_j Y_j(e) \). We have \( \mathbb{E}[Y] \leq c_e/(16d) \). Now, for any \( i \),

\[
\Pr[Z_i(e) = 0] = \Pr \left[ \sum_{j=1}^{i-1} \rho_j Y_j(e) > c_e - \rho_i \right] \\
\leq \Pr[Y > c_e - 1].
\]

Let \( \beta = (c_e - 1)/\mathbb{E}[Y] \). The Chernoff–Hoeffding bound, after some routine algebra, gives

\[
\Pr[Y > c_e - 1] \leq \left( \frac{e^\beta}{\beta^\beta} \right)^{\beta^{-1} c_e - 1} \\
\leq \left( \frac{e}{\beta} \right)^{c_e - 1} \\
\leq \left( \frac{ec_e}{16d(c_e - 1)} \right)^{c_e - 1} \\
\leq \left( \frac{1}{2} \right)^{1/n^2}
\]

for large enough \( d \). Finally, recall that we started with a fractional solution that was \((\log n, d)\)-favorable, i.e., on any path \( \pi \) used by the solution, there are at most \( d \) edges of “small” capacity \((\leq \log n)\). Applying (3) for the small-capacity edges on \( \pi \) and (4) for the large-capacity edges yields

\[
\sum_{e \in \pi} \Pr[Z_i(e) = 0] \leq d \cdot \frac{2 + 2e}{16d} + n \cdot \frac{1}{n^2} \leq \frac{1}{2}.
\]

Using this in (2) gives \( \Pr[A_i(\pi) = 1] \geq \Pr[X_i(\pi) = 1] \cdot (1 - \frac{1}{2}) = f_{\pi}/(32d) \).

As the selection phase chooses at most one path for each \( i \), we have \( \Pr[A_i = 1] = \sum_{\pi \in \mathcal{P}_i} \Pr[A_i(\pi) = 1] \geq \sum_{\pi \in \mathcal{P}_i} f_{\pi}/(32d) = x_i/(32d) \). Therefore the expected profit of the final solution (after the pruning phase) is \( \sum_{i=1}^k w_i \Pr[A_i = 1] \geq \sum_{i=1}^k w_i x_i/(32d) = W/(32d) = \Omega(W/d) \).

This completes the proof of Part 1 of Theorem 3.5.
3.2.2. Exploiting a gap between demands and capacities. We now complete the proof of Theorem 3.5 by proving the remaining two parts.

PROOF OF THEOREM 3.5 (Part 2). So far we have worked with arbitrary demands subject only to the no-bottleneck assumption. This part of the theorem applies when we have the stronger guarantee that there is a separation between the largest demand and the smallest capacity (i.e., in the language of our normalized variables, $\rho_{\text{max}}$ is bounded away from 1). For the proof, we modify the above rounding algorithm to produce even better solutions.

Suppose $\rho_{\text{max}} \leq 1/B$ for some integer $B \geq 3$. We run the rounding algorithm just as above, except that in the selection phase we select a path $\pi$ with probability $f_{\pi} / (3e^{d_1/(B-1)})$, instead of the earlier $f_{\pi} / (16d)$. We define the random variables $X_i(\pi)$, $Y_i(e)$, $Z_i(e)$, $A_i(\pi)$, and $A_j$ as in Section 3.2.1 and analyze our rounding procedure just as before, except that instead of (3) we use the following inequality:

$$\Pr[Z_i(e) = 0] \leq e^{B \left( \frac{1}{3e^{d_1/(B-1)}} \right)^{B-1}} \leq \frac{1}{3d},$$

which we obtain from Part 2 of Lemma 3.8 using $\theta = 1/(3e^{d_1/(B-1)})$. Continuing the analysis as before, we eventually obtain $\sum_{e \in \pi} \Pr[Z_i(e) = 0] \leq \frac{1}{2}$, which gives us an expected profit of at least $W/(6e^{d_1/(B-1)}) = \Omega(W/d^{1/(B-1)})$. Finally, for the case $B = 2$ (when all demands lie in the range $[0, \frac{1}{2}]$), the bound $\Omega(W/d^{1/(2-1)}) = \Omega(W/d)$, which we already showed in Section 3.2.1.

PROOF OF THEOREM 3.5 (Part 3). This part of the theorem handles the case when we have an even stronger guarantee on the demands: they are discrete, i.e., each $\rho_i = 1/B$ for some integer $B \geq 2$. For this case we assume that the edge capacities are integers: if the capacities are not integral, we can round the capacity $c_e$ of each edge down to $\lfloor c_e \rfloor$, and scale down the fractional solution (by at most a factor of two) to maintain the feasibility of the solution. Now we use a rounding algorithm where we select a path $\pi$ with probability $f_{\pi} / (2ed^{1/B})$. The analysis is very similar to the ones above, but since $a_i(e) = \rho_i/c_e = 1/c_e B$ (and $c_e$ is a positive integer), we can use Part 3 of Lemma 3.8; indeed, setting $\theta = 1/(2ed^{1/B})$, we obtain the following:

$$\Pr[Z_i(e) = 0] \leq e^{B \left( \frac{1}{2ed^{1/B}} \right)^B} \leq \frac{1}{4d}.$$  

Continuing the analysis as before, we see that our expected profit is at least $W/(4ed^{1/B}) = \Omega(W/d^{1/B})$. Finally, note that the claimed guarantee for the case of $B = 1$ is $\Omega(W/d^1)$ which follows from Section 3.2.1, and hence our result for the discrete case holds for all $B \geq 1$.

4. An Improved Result for Strong Expanders. For strong expanders as specified in Definition 2.2, we can improve the results in Corollary 3.6 to obtain the following theorem.
THEOREM 4.1. For sufficiently large constant $\Delta$, there is an approximation algorithm with ratio $O(\sqrt{\log n})$ for UCUFP on $\Delta$-regular strong expanders.

To prove this, we need the following result of Frieze [17].

THEOREM 4.2 (Frieze). There exist constants $k_1, k_2$ such that given an $n$-vertex $\Delta$-regular strong expander, with $\Delta$ sufficiently large, any $(k_1 \Delta n / \log n)$ pairs of vertices, with no vertex appearing in more than $k_2 \Delta$ pairs, can be connected by disjoint paths of length $O(\log n)$ in polynomial time.

In order to prove the theorem above, we need the following lemma.

LEMMA 4.3. Given a graph $G = (V, E)$ with weights $w_e$ on edges, and parameters $k$ and $C$, suppose we want to find a set of $k$ (or fewer) edges of maximum total weight such that no vertex is adjacent to more than $C$ of these edges. There is a polynomial time $O(1)$ approximation algorithm for this problem. In fact, the approximation ratio of this algorithm remains a constant even if we allow the optimum to have $C'$ edges adjacent to any vertex, where $C'$ is $O(C)$.

PROOF OF LEMMA 4.3. Consider the greedy algorithm that repeatedly picks the heaviest unpicked edge that does not already have $C$ picked edges incident to one of its endpoints. The algorithm stops if $k$ edges have been picked or if there are no edges that can be picked. We claim that this algorithm is a constant factor approximation algorithm for this problem. Let $F$ be the set of edges picked by our algorithm, and let $F^*$ be the set of edges in an optimal solution.

We only need to bound the cost of the edges in $F^* \setminus F$. Let $V'$ be the set of vertices $v$ such that $F$ has $C$ edges incident to $v$. For $v \in V'$, let $l_v$ be the smallest weight of an edge in $F$ incident to $v$. It is easy to see that $\sum_{v \in V'} C \cdot l_v$ is at most twice the weight of the edges in $F$. We claim that the weight of the edges in $F^* \setminus F$ is at most $\sum_{v \in V'} C \cdot l_v$.

Indeed, let $e \in F^* \setminus F$. When the greedy algorithm considers $e$ but does not pick it, our solution must have already picked $C$ edges incident to one of the endpoints of $e$. Let this endpoint be $u$. Then $w_e \leq l_u$ and we charge the weight of $e$ to $u$. Since $F^*$ contains at most $C$ edges incident to any vertex, this charging scheme charges at most $C \cdot l_v$ to any vertex in $v$ in $V'$. Hence the total weight of edges in $F^* \setminus F$ is at most $\sum_{v \in V'} C \cdot l_v$. The second part of our lemma is also easy to see, because then the total charge on any vertex $v$ is at most $C' \cdot l_v$.

PROOF OF THEOREM 4.1. Suppose we have an instance $I$ of UCUFP. Fix an optimal integral solution $O$ for $I$ and partition the terminals pairs of $O$ into three parts as follows:

- $O_1$ includes exactly those pairs with demand at most $\frac{1}{2}$.
- $O_2$ includes exactly those pairs not in $O_1$, and routed by $O$ on paths of length at most $\sqrt{\log n}$.
- $O_3$ includes the rest of the pairs.

We use these three parts to prove the performance guarantee of our algorithm, which we now describe.
Our algorithm partitions \( \mathcal{I} \) into two instances: \( \mathcal{I}_1 \), which is \( \mathcal{I} \) restricted to demands \( \rho_i \leq \frac{1}{2} \), and \( \mathcal{I}_2 \), which is \( \mathcal{I} \setminus \mathcal{I}_1 \). By Corollary 3.6, applied with \( B = 2 \), we can find a solution to \( \mathcal{I}_1 \) that \( O(\sqrt{\Delta \alpha^{-1} \log n}) \)-approximates the optimum (here \( \alpha \) is as in Definition 2.2). Since \( \mathcal{O}_1 \) is a feasible solution for \( \mathcal{I}_1 \), our solution is within the same factor of \( \mathcal{O}_1 \).

Now we solve an LP relaxation of \( \mathcal{I}_2 \) with the added restriction that flow path lengths are at most \( \sqrt{\log n} \). By Part 1 of Theorem 3.5, the fractional solution can be rounded to give an \( O(\sqrt{\log n}) \) approximation to the LP optimum. Since \( \mathcal{O}_2 \) is feasible for the LP relaxation, we obtain an \( O(\sqrt{\log n}) \) approximation to the value of \( \mathcal{O}_2 \).

Finally, we want to get an approximation to \( \mathcal{O}_3 \). First we bound \( |\mathcal{O}_3| \). Each demand in \( \mathcal{O}_3 \) is more than \( \frac{1}{2} \), thus a feasible solution induces edge disjoint paths for the pairs. Further, by definition, each pair in \( \mathcal{O}_3 \) uses a path of length at least \( \sqrt{\log n} \). Therefore, it follows that \( |\mathcal{O}_3| \leq |E(G)|/\sqrt{\log n} = O(\Delta n/\sqrt{\log n}) \). We choose a set of pairs \( S \) such that \( |S| \leq k_1 \Delta n/\sqrt{\log n} \) and no more than \( k_2 \Delta \) pairs in \( S \) are incident to any vertex. Here \( k_1 \) and \( k_2 \) are the constants in Theorem 4.2. We do this as follows. We build an auxiliary graph \( G' \) on the vertex set \( V \); for each demand \((s_i, t_i)\) in \( \mathcal{I}_2 \) we have an edge in \( G' \) between \( s_i \) and \( t_i \) and the weight of this edge is \( w_i \). Now, we obtain \( S \) as the collection of edges in \( G' \) using Lemma 4.3 with \( k = (k_1 \Delta n/\log n) \), \( C = k_2 \Delta \), and \( C' = \Delta \). From the lemma, the total weight of the pairs in \( S \) is within a constant factor of the \( k_1 \Delta n/\sqrt{\log n} \) pairs in \( \mathcal{O}_3 \) of largest weight. Since \( |\mathcal{O}_3| = O(\Delta n/\sqrt{\log n}) \), it follows that the total weight of pairs in \( S \) is within an \( \Omega(1/\sqrt{\log n}) \) factor of the total weight of pairs in \( \mathcal{O}_3 \). Theorem 4.2 guarantees that all the pairs in \( S \) can be routed via edge disjoint paths and hence we obtain an \( O(\sqrt{\log n}) \) approximation to \( \mathcal{O}_3 \).

Since one of \( \mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3 \) has at least a third of the profit of \( \mathcal{O} \) and we approximated each within an \( O(\sqrt{\log n}) \) factor, we get the desired result. \( \square \)

### 5. Line and Ring Networks

In this section we consider UFP restricted to the line network. We handle the ring network in a very similar fashion; we give the relevant details at the end of the section. Before we proceed, we fix some notation. The terminal pairs now form intervals \( I_1, I_2, \ldots, I_m \) on the line \([1, n] \), with \( I_j \) having demand \( \rho_j \) and weight (or profit) \( w_j \). Edge \( e \) on the line has capacity \( c(e) \). For an edge \( e \), let \( \mathcal{I}(e) \) be the set of all demands (intervals) that contain \( e \). Recall that we are working under the no-bottleneck assumption: \( \rho_{\text{max}} \leq 1 = c_{\text{min}} \).

The UCUFP on the line is equivalent to a resource allocation problem that has been studied recently [29], [7], [10], [13]; however, we do not use the resource allocation terminology. Constant factor approximation algorithms for the resource allocation problem, and consequently UCUFP on the line, have been obtained via several different techniques—LP rounding [8], [29], the local-ratio method [7], [10], and primal-dual algorithms [7], [10]. Most of these techniques do not seem to extend to UFP on the line where capacities are non-uniform. There is, however, one exception: a recent algorithm of Calinescu et al. [13] which gives constant factor approximations for UCUFP on the line. We extend their algorithm and analysis to non-uniform capacities. Their algorithm is the following: the demands are divided into two sets, one set containing demands which are “large” compared with the (common) capacity, say 1, and the other containing the rest. Dynamic programming is then invoked to find the optimal solution on the set of
large demands. For the “small” demands, the algorithm solves the LP and then randomly rounds the solution (after scaling it by a constant $\alpha < 1$). The resulting set of demands has the right weight in expectation, but it may not be feasible. The alteration step then looks at the randomly chosen demands in order of their left endpoints, accepting a demand in the final output if adding it maintains feasibility. Since all edges have capacity 1, this is too weak to give a constant factor approximation and we need a more careful analysis. The alteration step then allows us to make the concentration arguments. To this end, we define the bottleneck edge for some $\delta$-large demand $I_j$ to be the capacity of the lowest capacity edge on this demand. Now a demand $I_j$ is $\delta$-small if $\rho_j \leq \delta b_j$, else it is $\delta$-large.

In what follows, we show how to find the optimal solution for the $\delta$-large demands, and a constant factor approximation for the set of $\delta$-small demands, for some appropriate choice of $\delta$. We then output the better of the two solutions.

5.1. The Large Demands. The following lemma is key to invoking dynamic programming to find an optimal solution for the $\delta$-large demands in $n^{O(1/\delta^2)}$ time.

**Lemma 5.1.** The number of $\delta$-large demands that cross an edge in any feasible solution is at most $2\lfloor 1/\delta^2 \rfloor$.

**Proof.** Fix a feasible solution $S$, and consider an edge $e$. Let $S_e$ be the set of all $\delta$-large demands in $S$ that cross $e$. We partition $S_e$ into two sets $S_e^l$ and $S_e^c$ as follows: a demand in $S_e^c$ is in $S_e$ if it has a bottleneck capacity edge to the left of $e$ (including $e$), otherwise the demand is in $S_e^l$. We show that $|S_e^c| \leq \lfloor 1/\delta^2 \rfloor$, and a similar argument shows that $|S_e^l| \leq \lfloor 1/\delta^2 \rfloor$.

Let $A$ be the set of bottleneck edges for demands in $S_e^l$ and let $e'$ be the rightmost edge in $A$. Since $e'$ is the bottleneck edge for some $\delta$-large demand $I_j \in S_e^l$, by definition, $\rho_j \geq \delta c(e')$. Since $\rho_j \leq c_{\min}$, it follows that $c(e') \leq c_{\min}/\delta$. Because $e'$ is the rightmost edge in $A$, all demands in $S_e^l$ pass through $e'$. However, each demand $I_k$ in $S_e^l$ is $\delta$-large, which implies that $\rho_k \geq \delta b_k \geq \delta c_{\min}$. It follows that $|S_e^l| \leq \lfloor c(e')/(\delta c_{\min}) \rfloor \leq \lfloor 1/\delta^2 \rfloor$.

Using Lemma 5.1 and standard dynamic programming ideas, we obtain the following:

**Theorem 5.2.** For an instance of UFP on a line network that has only $\delta$-large demands, an optimum solution can be found in $n^{O(1/\delta^2)}$ time.
The dynamic program computes a table \( T[i, S] \), for all values of \( i \) between 1 and \( n - 1 \), and for all subsets \( S \) of demands that contain the edge \((i, i + 1)\) and such that \( |S| \leq 2|1/\delta^2| \). The table entry \( T[i, S] \) is the maximum profit that can be achieved from demands with their left endpoints in \( 1, 2, \ldots, i \) and such that \( S \) is the subset of demands among them that contain the edge \((i, i + 1)\). It is easy to see that the table entry \( T[i, S] \) can be computed if all entries of the form \( T[i', \cdot] \) are available for \( 1 \leq i' < i \). Thus \( T \) can be computed sequentially in increasing order of \( i \).

5.2. The Small Demands. We now show that for any constant \( \delta < \frac{1}{2} \), when all demands in a UFP instance are \( \delta \)-small, we can give an \( O(1) \) approximation to the optimal solution. The approximation factor deteriorates as \( \delta \) increases. On the other hand, the running time of the algorithm in Theorem 5.2 increases as \( \delta \) decreases. We choose the parameters as follows:

\[
\delta = 0.001 \quad \text{and} \quad \alpha = 0.032.
\]

We first solve the linear program \( \text{LPMAIN} \) for the problem. Let \( x_j \) be the fractional value assigned to demand \( I_j \). We define two \( \{0, 1\} \)-random variables \( X_j \) and \( Y_j \) as follows:

1. Let \( X_j \) be set to 1 independently with probability \( \alpha x_j \).
2. Sort the demands corresponding to \( X_j = 1 \) in order of their left endpoints (breaking ties arbitrarily). Consider them in this order, adding the current demand to the output if the addition does not violate any edge capacity. Set \( Y_j = 1 \) if demand \( I_j \) is output.

By construction, this procedure produces a feasible solution. Clearly, \( E[X_j] = \Pr[X_j = 1] = \alpha x_j \). The probability that \( I_j \) will be in the final solution is \( E[Y_j] = \Pr[Y_j = 1] = \alpha x_j \cdot \Pr[Y_j = 1 \mid X_j = 1] \). The rest of the argument shows that \( \Pr[Y_j = 0 \mid X_j = 1] \), the chance of rejection, is at most 0.597; this, in turn, shows that the expected weight of the solution is at least \( \sum_j w_j x_j / 77.51 \), i.e., a constant factor away from the weight of the fractional solution.

We focus on a particular demand \( I_j \) with \( X_j = 1 \), and we let \( E_j = (e_{i_1}, \ldots, e_{i_k}) \) be the edges on \( I_j \) from left to right. The crucial idea is the following: when considering \( I_j \), its probability of rejection depends on whether there is "enough room" on all these edges. Instead of taking a union bound over all edges, we choose a subsequence of edges such that the capacity of each edge drops by half, and such that for a demand to be rejected, a "bad" event happens at one of these chosen edges. Now a union bound on the bad events at these edges suffices. We show that this union bound gives us a sum whose terms decrease rapidly—faster than geometrically—and thus the chance of rejection is a constant times the probability of rejection on some edge \( e_i \). Finally, arguments similar to that in [13] complete the proof.

Formally, create a subsequence \( E_j' = (e_{i_1}, e_{i_2}, \ldots, e_{i_k}) \) of \( E_j \) as follows: set \( i_1 = 1 \), and hence \( e_{i_1} = e_1 \). For \( t > 1 \), set \( i_t = \min\{t: \ t > i_{t-1} \text{ and } c(e_t) < \alpha c(e_{i_{t-1}})/2\} \). In other words \( e_{i_t} \) is the leftmost to the right of \( e_{i_{t-1}} \), with capacity at most half the capacity of \( e_{i_{t-1}} \). If there is no such edge we stop the construction of the sequence. For \( 1 \leq a \leq h \), let \( E_a \) denote the (bad) event that the random demands chosen in step 1 use at least \( \frac{1}{4} c(e_{i_a}) - \delta b_j \) capacity in the edge \( e_{i_a} \). Recall that \( b_j \) is the bottleneck capacity of \( I_j \). The following lemma shows that it is enough to bound the chance that no bad event occurs on these chosen edges.
LEMMA 5.3. \( \Pr[Y_j = 0 \mid X_j = 1] \leq \sum_{a=1}^{h} \Pr[E_a] \).

PROOF. If \( Y_j = 0 \) and \( X_j = 1 \) then some edge \( e_x \in E_j \) had a capacity violation when \( I_j \) was considered for insertion. Let \( e_{i_j} \) be the edge in \( E'_j \) to the left of \( e_x \) and closest to it. (Here, an edge is considered to be “to the left” of itself.) Note that such an edge always exists since \( e_{i_1} = e_1 \), and \( e_1 \) is the leftmost edge in \( I_j \). By the construction of the subsequence, \( c(e_x) \geq \frac{1}{2} c(e_{i_j}) \). If the capacity of \( e_x \) was violated while trying to insert \( I_j \), it must be that the capacity of demands already accepted that cross \( e_x \) is at least \( c(e_x) - \rho_j \) which is lower bounded by \( \frac{1}{2} c(e_{i_j}) - \delta b_j \); we use the fact that \( I_j \) is small which implies that \( \rho_j \leq \delta b_j \) and the fact that \( c(e_x) \geq \frac{1}{2} c(e_{i_j}) \). However, any interval that is accepted before \( I_j \) and crosses \( e_x \), must also cross \( e_{i_j} \), and thus event \( E_a \) occurs. Applying the trivial union bound, we have \( \Pr[Y_j = 0 \mid X_j = 1] \leq \sum_{a=1}^{h} \Pr[E_a] \). \( \square \)

It is not enough to bound each \( \Pr[E_a] \) by a constant, because we may have to take a union bound over up to \( \Theta(n) \) of these. However, the following lemma addresses this concern.

LEMMA 5.4. For our particular choices of \( \alpha \) and \( \delta \), we have \( \Pr[E_a] \leq (0.4051)^{c(e_a)} \), and, therefore, \( \sum_{a=1}^{h} \Pr[E_a] \leq 0.597 \).

PROOF. Let \( Q_a = \sum_{t \in \mathcal{I}(e_a)} \rho_t X_t \) be the random variable that gives the sum of demands that edge \( a \) intersects and that are chosen in step 1. Since each \( \rho_t \leq 1 \), the independent variables \( \{\rho_t X_t\} \) are distributed in \([0, 1]\). We have \( \Pr[E_a] = \Pr[Q_a \geq \frac{1}{2} c(e_{i_a}) - \delta b_j] \). Setting \( \beta = (\frac{1}{2} - \delta - \alpha) / \alpha \), and using the fact that \( b_j \leq c(e_{i_a}) \) gives

\[
\Pr[E_a] = \Pr[Q_a \geq \frac{1}{2} c(e_{i_a}) - \delta b_j] \leq \Pr[Q_a \geq (1 + \beta) \alpha c(e_{i_a})].
\]

Also,

\[
E[Q_a] = \sum_{t \in \mathcal{I}(e_a)} \rho_t E[X_t] = \sum_{t \in \mathcal{I}(e_a)} \alpha \rho_t x_t \leq \alpha c(e_{i_a}),
\]

where the last inequality follows from the feasibility of the LP solution. Since \( Q_a \) is a sum of independent random variables distributed in \([0, 1]\) we apply a Chernoff–Hoeffding bound to get

\[
\Pr[Q_a \geq (1 + \beta) \alpha c(e_{i_a})] \leq (e^\beta / (1 + \beta)^{1+\beta})^{\alpha c(e_{i_a})} \leq (0.4051)^{c(e_{i_a})},
\]

where the final inequality follows by plugging in the constants we chose for \( \alpha \) and \( \delta \). Since \( c(e_{i_a}) < c(e_{i_{a-1}}) / 2 \) and each \( c(e_{i_a}) \geq 1 \), we now get

\[
\sum_{a=1}^{h} \Pr[E_a] \leq \sum_{a} (0.4051)^{c(e_a)} \leq \sum_{i \geq 0} (0.4051)^2 \leq 0.597,
\]

which proves the lemma. \( \square \)

The previous two lemmas together imply \( \Pr[Y_j = 0 \mid X_j = 1] \leq 0.597 \), and so the approximation ratio of our algorithm is at most \( 1 / (0.403 \alpha) \leq 77.51 \).
5.3. Combining Large and Small Demands. We combine the algorithms for large and small demands in a straightforward manner. Partition any given instance of UFP on the line into two sub-instances: $I_L$, which contains only the $\delta$-large demands, and $I_S$, which contains only the $\delta$-small demands. Solve $I_L$ optimally, and find a $77.51$ approximation to the optimum of $I_S$; we know how to do both these things in polynomial time. Then simply output the better of the two solutions.

In an optimal solution to $I$, either the small demands contribute at least a $77.51/78.51$ fraction of the weight, or the large demands contribute at least a $1/78.51$ fraction of the weight. Choosing the better of the two solutions, as above, ensures that we always obtain at least a $1/78.51$ fraction of the weight of an optimal solution to $I$. Thus, we have proved the following theorem (we remind the reader that we have not tried to optimize our constants).

**Theorem 5.5.** There is a polynomial time $78.51$ approximation for UFP on the line if $\rho_{\text{max}} \leq \epsilon_{\text{min}}$.

**Corollary 5.6.** There is a constant factor approximation for UFP on the line when $\rho_{\text{max}} / \rho_{\text{min}}$ is bounded even without the no-bottleneck assumption. Hence, for arbitrary demands we get an $O\left(\log(\rho_{\text{max}} / \rho_{\text{min}})\right)$ approximation.

**Proof Sketch.** Since the analysis for the $\delta$-small demands does not use the fact that $\rho_{\text{max}} \leq \epsilon_{\text{min}}$, we need only consider the large demands. For the $\delta$-large demands, an argument similar to that in Lemma 5.1 and Theorem 5.2 works when $\rho_{\text{max}} / \rho_{\text{min}}$ is bounded.

So, given arbitrary demands, we can divide them into $O\left(\log(\rho_{\text{max}} / \rho_{\text{min}})\right)$ classes so that for any two demands $j$ and $j'$ in the same class, $\rho_j / \rho_{j'}$ is bounded. This proves the result.

5.4. Integrality Gap. In this section we show that the integrality gap of the natural LP for instances with $\rho_{\text{max}} \leq \epsilon_{\text{min}}$ is upper bounded by some fixed constant. The algorithm described in the previous section uses dynamic programming for large demands and hence it does not imply a constant factor bound on the integrality gap of the LP. If we do not have the no-bottleneck assumption, the integrality gap can be $\Theta(\log(\rho_{\text{max}} / \rho_{\text{min}}))$. Thus, the performances of our algorithms, both with and without the no-bottleneck assumption, match the integrality gap of the LP to within a constant factor.

**Theorem 5.7.** The integrality gap of the natural LP is $O(1)$ when $\rho_{\text{max}} \leq \epsilon_{\text{min}}$. For arbitrary demands the integrality gap is $\Theta(\log(\rho_{\text{max}} / \rho_{\text{min}}))$ which can be $\Omega(n)$.

**Proof.** We first show that the integrality gap of the natural LP is $O(1)$ when $\rho_{\text{max}} \leq \epsilon_{\text{min}}$. Consider a fractional solution to LPMAIN. Let $x_j$ be the fractional value assigned to demand $I_j$. Let $S_\delta$ denote the set of $\delta$-small demands, where $\delta$ is as chosen in Section 5.2. Consider the fractional profit accrued by the $\delta$-small demands, i.e., $\sum_{j \in S_\delta} w_j x_j$. If this is at least half the total profit of the LP solution, then we are done. This is so because
we have already shown that the LP restricted to \( \delta \)-small demands, for sufficiently small \( \delta \), has at most a constant integrality gap.

Let \( L_\delta \) be the set of demands \( i \) such that \( \rho_i \geq \delta \rho_{\max} \). Note that \( L_\delta \) contains the \( \delta \)-large demands but could include some \( \delta \)-small demands as well. If \( S_\delta \) does not have half the profit of the LP solution then clearly \( L_\delta \) does. We show how we can obtain an \( \Omega(\delta) \) fraction of the profit of \( L_\delta \). Suppose, for each demand \( i \in L_\delta \), we increase \( \rho_i \) to \( \rho_i' = \rho_{\max} \) and decrease \( x_i \) to \( x_i' = \delta x_i / 2 \). Also, for each edge \( e \), we round down the capacity \( c(e) \) to \( c'(e) \) such that \( c'(e) \) is the largest integer multiple of \( \rho_{\max} \) less than \( c(e) \)—note that \( c'(e) \) is at least \( c(e) / 2 \). From this, it is easy to see that \( x' \) is a feasible solution to \( \rho' \) with edge capacities \( c' \). The profit of this solution has decreased by at most \( \delta / 2 \). Also, any feasible solution to this new instance is feasible for the original instance since, in going from the original to the new instance, we only increased demands and reduced capacities. Observe that the instance we have created has all demands of equal size and all capacities that are integer multiples of the demand. For such instances, which are basically unit demand instances, the integrality gap of the LP is 1. This is a well-known fact and follows from the total unimodularity of consecutive-ones matrices [32]. Therefore we can recover a \( \delta / 2 \) fraction of the profit of \( L_\delta \).

Thus, for some sufficiently small but fixed \( \delta \), either \( S_\delta \) or \( L_\delta \) gives an \( f(\delta) \) fraction of its fractional profit for some function \( f \). It follows that the integrality gap of the LP is \( O(1) \). We now remove the assumption that \( \rho_{\max} \leq c_{\min} \). In this case we partition the demands into \( O(\log \rho_{\max} / \rho_{\min}) \) sets \( S_0, S_1, S_2, \ldots \), where \( S_j \) contains demands \( i \) such that \( \rho_i \in [\rho_{\max} / 2^j, \rho_{\max} / 2^{j+1}) \). For demands in any particular \( S_j \), using arguments similar to those in Corollary 5.6, we can show that the integrality gap of the LP is \( O(1) \). Picking the set \( S_j \) with the largest fractional profit from the LP shows that the integrality gap of the LP is \( O(\log \rho_{\max} / \rho_{\min}) \).

We now prove that the integrality gap is \( \Omega(\log(\rho_{\max} / \rho_{\min})) \). Consider the line graph on \( n + 1 \) points corresponding to the integers in \([0, n] \). The capacity of the edge \((i, i + 1)\) is \( 1 / 2^j \). We have \( n \) demands. Demand \( I_j \) corresponds to the interval \([0, j]\), and \( \rho_j = 1 / 2^{j-1} \). All demands have profit 1. We claim that any integral solution can route only one demand. To see this, let \( I_j \) be the demand with the smallest index that is routed in the solution. \( I_j \) saturates the edge \((j - 1, j)\) and hence no other demand \( I_{j'} \), \( j' > j \) can be routed. Thus, any integral solution has profit at most 1. Now we construct a fractional solution with profit \( \Omega(n) \). The fractional solution assigns \( x_j = \frac{1}{2} \) for all demands \( I_j \). Consider the edge \((j, j + 1)\). The demands which contain this edge are \( I_j, j' > j \). However, note that \( \sum_{j': j' > j} \rho_{j'} \leq 2 / 2^j \). Thus, the fractional solution is a feasible solution. The profit of this solution is \( n / 2 \). Also, notice that \( \rho_{\max} / \rho_{\min} = O(2^n) \). Thus, we have shown the desired integrality gap.

A bound of \( \Theta(\log \rho_{\max} / c_{\min}) \) on the integrality gap can also be obtained by slightly altering the above proof. We thank a reviewer for pointing this out.

5.5. **UFP on a Ring Network.** Finally, we consider UFP on the ring network. Unlike the line network, this gives us a choice of one of two paths for each demand. However, we can reduce the problem on the ring to that on a line network with a slight loss in the approximation factor as follows. Let \( e \) be any edge on the ring with \( c(e) = c_{\min} \). Consider any integral optimal solution \( O \) to the problem. The demands routed in \( O \) can
be partitioned into two sets $O_1$ and $O_2$ where those in $O_1$ use $e$ and those in $O_2$ do not. We remove $e$ and solve the problem approximately on the resulting line network. This clearly approximates the value of $O_2$. To approximate the solution for $O_1$, for each demand we choose the path that uses $e$ and solve a knapsack problem to find a $(1 + \varepsilon)$ approximation to the maximum weight set of demands that can be routed with capacity bounded by $c_e$, where $\varepsilon$ is an arbitrarily small constant. Since $c(e) = c_{min}$, any solution feasible at $e$ will be feasible for the entire network. Thus we obtain:

**THEOREM 5.8.** For UFP on the ring there is a $(1 + A + \varepsilon)$ approximation where $A$ is the approximation factor for the problem on the line, and $\varepsilon > 0$ is any fixed constant.

6. Concluding Remarks. Our $O(F_G \log n)$ approximation algorithm for UFP is based on randomly rounding a fractional solution to the LP relaxation. Two natural questions suggest themselves: namely, (a) whether there exists a deterministic algorithm with a similar approximation ratio, and (b) whether there exists an online algorithm with a similar competitive ratio. The answer to both these questions is positive. For the former, note that the randomized algorithm presented in this paper can be derandomized using the method of conditional expectations and pessimistic estimators [31]. The standard details are not particularly illuminating, and hence we omit them.

Perhaps of greater interest is an online algorithm with a competitive ratio of $O(F_G \log n)$ for the case of the throughput measure, that is, the case where the weights are proportional to the request size ($w_i = \rho_i$). Such an algorithm can be obtained by combining the bounded greedy algorithm [21], [25] and the algorithm of Awerbuch, Azar, and Plotkin (AAP) [5] for large capacities. Kleinberg [21] has previously developed and analyzed such a combined algorithm in a related context. However, to apply this idea in our context, we need the existence of $(\log n, O(F_G \log n))$-favorable solutions to the LP. We briefly describe our algorithm here. An edge is called a low-capacity edge if its capacity is less than $\log n$, and is called high-capacity otherwise. The AAP algorithm assumes that all edges are of high capacity; it maintains edge lengths that are exponential in the congestion of the edge—recall that the congestion of an edge $e$ is the flow already routed on $e$, divided by the capacity of $e$. A path is good for the AAP algorithm if the total length of the path is at most some given bound (the $L_{AAP}$ bound) that depends only on $n$, the graph size. (Since we are only offering a sketch of the extension, we omit the precise definition of the AAP factor here.)

The bounded greedy algorithm (BGA) [21], [25] is relevant for the low-capacity case. A path is good for the BGA if the number of edges in it is at most a given bound $B$. We combine these two measures as follows. In the combined algorithm, we call a path good if the total number of low-capacity edges in it is $O(F_G \log n)$, and the total length of the high-capacity edges is less than the AAP bound $L_{AAP}$. The online algorithm works as follows. When a new pair $(u, v)$ arrives, if a feasible good path exists between $u$ and $v$, we route the demand pair $(u, v)$ along any such path, otherwise we reject it. The lengths of the high-capacity edges are updated according to the AAP algorithm. The claimed competitive ratio can be obtained by combining the analysis for the bounded greedy algorithm from [25] with that of [5] and using the existence of a $(\log n, O(F_G \log n))$-favorable solution to the LP. We note that the idea of combining
the algorithm (and analysis) for high and low capacity edges is borrowed from earlier work of Kleinberg [21].

For UFP on the line and the ring, a $(2 + \varepsilon)$ approximation has been obtained in subsequent work by Chekuri et al. [15]. The improvement is based on a different algorithm for small demands which builds on certain grouping and scaling ideas for packing problems from the work of Kolliopoulos and Stein [23].

Acknowledgments. We are grateful to Bruce Shepherd for suggesting the unsplittable flow problem on the line and for several discussions. We thank Petr Kolman for some clarifications on the results in [25] and Thomas Erlebach for pointing out an error in Lemma 5.4 in an earlier version of the paper. We are grateful to a referee for a thorough and careful reading of the paper which helped improve its presentation.

Appendix. The Technical Probabilistic Lemma. We give here a proof of the probabilistic lemma that we used to analyze the rounding-and-alteration algorithm of Section 3.2.

Lemma A.1 (Restatement of Lemma 3.8). Let $a_1, \ldots, a_h, y_1, \ldots, y_h \in [0, 1]$ be such that $\forall i < j: (a_i \leq \frac{1}{2} \Rightarrow a_j \leq \frac{1}{2})$ and furthermore $\sum_{i=1}^{h} a_i y_i \leq 1$. Let $0 < \theta < 1$ and let independent 0-1 random variables $Y_1, \ldots, Y_h$ and (possibly dependent) 0-1 random variables $Z_1, \ldots, Z_h$ be defined as follows:

$$Y_i = \begin{cases} 1, & \text{with probability } \theta y_i, \\ 0, & \text{otherwise}, \end{cases}$$

$$Z_i = \begin{cases} 1, & \text{if } \sum_{j<i} a_j Y_j \leq 1 - a_i \\ 0, & \text{otherwise}. \end{cases}$$

Then, for each $i$, $\Pr[Z_i = 0]$ is at most:

1. $(2 + 2e)\theta$, in general.
2. $e^B\theta^{B-1}$, if each $a_i \in [0, 1/B]$ for an integer $B \geq 2$.
3. $e^B\theta^B$, if each $a_i = 1/B$ for an integer $B \geq 1$.

Proof of Part 1. Fix an $i$ and define the index sets $I = \{j < i: a_j > \frac{1}{2}\}$ and $J = \{j < i: a_j \leq \frac{1}{2}\}$. Then we have

$$Z_i = 0 \Rightarrow (\exists j \in I: Y_j = 1) \lor \left(\sum_{j \in J} a_j Y_j > 1 - a_i \right).$$

Now

$$1 \geq \sum_{j=1}^{h} a_j y_j \geq \sum_{j \in I} a_j y_j > \sum_{j \in J} \frac{1}{2} y_j,$$

whence

$$\Pr[\exists j \in I: Y_j = 1] \leq \sum_{j \in I} \Pr[Y_j = 1] \leq \sum_{j \in J} \theta y_j < 2\theta.$$
Suppose $a_i > \frac{1}{2}$. By the condition on the $a_j$'s this means $J = \emptyset$. Using (7) and (8), we obtain $\Pr[Z_i = 0] < 2\theta$. On the other hand, suppose $a_i \leq \frac{1}{2}$. The random variables $\{2a_j Y_j\}_{j \in J}$ are independent and distributed in $[0, 1]$, and their sum $Y$ (say) satisfies $E[Y] = \sum_{j \in J} 2a_j \theta y_j \leq 2\theta$. Applying the Chernoff–Hoeffding bound,

$$
\Pr \left[ \sum_{j \in J} a_j Y_j > 1 - a_i \right] = \Pr[Y > 2 - 2a_i]
\leq \Pr[Y > 1]
\leq \left( \frac{e^{1/2\theta} - 1}{1/2\theta} \right)^{2\theta}
= 2\theta e^{1-2\theta}
\leq 2e\theta.
$$

Combining this with (8) and using (7), we get $\Pr[Z_i = 0] \leq (2 + 2e)\theta$.

**Proof of Part 2.** We first note that $I = \emptyset$ whence (7) simplifies to $\Pr[Z_i = 0] \leq \Pr[\sum_{j \in J} a_j Y_j > 1 - a_i]$. Now, the independent random variables $\{Ba_j Y_j\}_{j \in J}$ are distributed in $[0, 1]$ and their sum $Y$ (say) satisfies $E[Y] = \sum_{j \in J} B a_j \theta y_j \leq B\theta$. Applying the Chernoff–Hoeffding bound and arguing as above,

$$
\Pr \left[ \sum_{j \in J} a_j Y_j > 1 - a_i \right] = \Pr[Y > B - B a_i]
\leq \Pr[Y > B - 1]
\leq \left( \frac{e^{((B-1)/(B\theta))^{-1}} - 1}{(B-1)/(B\theta))^{(B-1)/(B\theta)}} \right)^{B\theta}
= \left( 1 + \frac{1}{B - 1} \right)^{B-1} . \theta^{B-1} e^{B-1-B\theta}
\leq e^{B\theta} B^{-1}.
$$

**Proof of Part 3.** We first consider the case $B \geq 2$. This works a lot like Part 2 above, except that the random variables $\{Ba_j Y_j\}_{j \in J}$ are now $\{0, 1\}$-variables, so their sum $Y$, if greater than $B - 1$, must be at least $B$. Thus,

$$
\Pr \left[ \sum_{j \in J} a_j Y_j > 1 - a_i \right] \leq \Pr[Y \geq B]
\leq \left( \frac{e^{1/\theta} - 1}{(1/\theta)^{1/\theta}} \right)^{B\theta}
= (\theta e^{1-\theta})^B
\leq e^B \theta^B,
$$
as desired. Finally, we consider the case \( B = 1 \). In this case every \( a_i = 1 \), so \( J = \emptyset \). Also, \( 1 \geq \sum_{j \in I} a_j y_j = \sum_{j \in I} y_j \), whence

\[
\Pr[\exists j \in I: Y_j = 1] \leq \sum_{j \in I} \Pr[Y_j = 1] \leq \sum_{j \in I} \theta y_j \leq \theta. \tag{9}
\]

Combining this with (7) gives us \( \Pr[Z_i = 0] \leq \theta \leq e^1 \theta^1 \), as desired.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
[2] Noga Alon and Joel Spencer. *The Probabilistic Method*. Wiley Interscience, New York, 1992.
[3] Matthew Andrews and Lisa Zhang. Hardness of the undirected edge-disjoint paths problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 276–283, 2005.
[4] Matthew Andrews, Julia Chuzhoy, Sanjeev Khanna, and Lisa Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 226–244, 2005.
[5] Baruch Awerbuch, Yossi Azar, and Serge Plotkin. Throughput-competitive online routing. In *Proceedings of the 34th Annual ACM Symposium on Foundations of Computer Science*, pages 32–40, 1993.
[6] Yossi Azar and Oded Regev. Strongly polynomial algorithms for the unsplittable flow problem. In *Proceedings of the 8th Integer Programming and Combinatorial Optimization Conference*, pages 15–29, 2001.
[7] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph (Seffi) Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *J. ACM*, 48(5):1069–1090, 2001. Preliminary version in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 735–744, 2000.
[8] Amotz Bar-Noy, Sudipto Guha, Joseph Naor, and Baruch Schieber. Approximating the throughput of multiple machines in real-time scheduling. *SIAM J. Comput.*, 31(2):331–352, 2001.
[9] Alok Baveja and Aravind Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Math. Oper. Res.*, 25(2):255–280, 2000.
[10] Piotr Berman and Bhaskar DasGupta. Improvements in throughput maximization for real-time scheduling. In *Proceedings of the 32nd Annual ACM symposium on Theory of Computing*, pages 680–687. ACM Press, New York, 2000.
[11] Tom Bohman and Alan Frieze. Arc-disjoint paths in expander digraphs. *SIAM J. Comput.*, 32(2):326–344, 2003.
[12] Andrei Z. Broder, Alan M. Frieze, and Eli Upfal. Existence and construction of edge-disjoint paths on expander graphs. *SIAM J. Comput.*, 23(5):976–989, 1994.
[13] Gruia Calinescu, Amit Chakrabarti, Howard Karloff, and Yuval Rabani. Improved approximation algorithms for resource allocation. In *Proceedings of the 9th Integer Programming and Combinatorial Optimization Conference*, pages 439–456. Volume 2337 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2001.
[14] Chandra Chekuri and Sanjeev Khanna. Edge disjoint paths revisited. In *Proceedings of the Fourteenth Annual ACM–SIAM Symposium on Discrete Algorithms* (Baltimore, MD, 2003), pages 628–637. ACM Press, New York, 2003.
[15] Chandra Chekuri, Marcelo Mydlarz, and F. Bruce Shepherd. Multicommodity demand flow in a tree (extended abstract). In *Automata, Languages and Programming*, pages 410–425. Volume 2719 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2003.
[16] Lisa K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discrete Math.*, 13(4):505–520, 2000.
[17] Alan M. Frieze. Edge-disjoint paths on expander graphs. *SIAM J. Comput.*, 30(6):1790–1801, 2001.
[18] Naveen Garg and Jochen Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, pages 300–309, 1998.

[19] Naveen Garg, Vijay Vazirani, and Mihalis Yannakakis. Primal–dual approximation algorithms for integral flow and multicut in trees. Algorithmica, 18(1):3–20, 1997. Preliminary version in Proceedings of the 20th International Colloquium on Automata, Languages, and Programming, pages 64–75, 1993.

[20] Venkatesan Guruswami, Sanjeev Khanna, Rajmohan Rajaraman, Bruce Shepherd, and Mihalis Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. J. Comput. System Sci., 67(3):473–496, 2003. Preliminary version in Proceedings of 31st ACM Symposium on Theory of Computing, pages 19–28, 1999.

[21] Jon M. Kleinberg. Approximation Algorithms for Disjoint Paths Problems. Ph.D. thesis, MIT, 1996.

[22] Jon M. Kleinberg and Ronitt Rubinfeld. Short paths in expander graphs. In Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, pages 86–95, 1996.

[23] Stavros Koliopoulos and Clifford Stein. Approximating disjoint-path problems using packing integer programs. Math. Programm. Ser. A, 99(1):63–87, 2004. Preliminary version in Proceedings of IPCO, pages 153–168, 1998.

[24] Petr Kolman and Christian Scheideler. Simple on-line algorithms for the maximum disjoint paths problem. Algorithmica, 39(3):209–233, 2004. Preliminary version in Proceedings of the 13th ACM Symposium on Parallel Algorithms and Architectures, pages 38–47, 2001.

[25] Petr Kolman and Christian Scheideler. Improved bounds for the unsplittable flow problem. To appear in J. Algorithms. Preliminary version in Proceedings of the 13th Annual ACM–SIAM Symposium on Discrete Algorithms, pages 184–193, 2002.

[26] F. Thomas Leighton and Satish B. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. J. ACM, 46(6):787–832, 1999. Preliminary version in Proceedings of the 29th Annual Symposium on Foundations of Computer Science, pages 422–431, 1988.

[27] Alex Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. Combinatorica, 8:261–277, 1988.

[28] Rajeev Motwani and Prabhakar Raghavan. Randomized Algorithms. Cambridge University Press, New York, 1995.

[29] Cynthia A. Phillips, R. N. Uma, and Joel Wein. Off-line admission control for general scheduling problems. J. Sched., 3(6):365–381, 2000. Preliminary version in Proceedings of 11th SODA, pages 879–888, 2000.

[30] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. Math. Oper. Res., 20(2):257–301, 1995.

[31] Prabhakar Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. J. Comput. System Sci., 37(2):130–143, 1988. Preliminary version in Proceedings of FOCS, pages 10–18, 1986.

[32] Alexander Schrijver. Theory of Linear and Integer Programming. Wiley, New York, 1986.

[33] Aravind Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, pages 416–425, 1997.

[34] Aravind Srinivasan. New approaches to covering and packing problems. In Proceedings of the 12th Annual ACM–SIAM Symposium on Discrete Algorithms, pages 567–576, 2001.

[35] Kasturi Varadarajan and Ganesh Venkataraman. Graph decomposition and the Greedy algorithm for edge-disjoint paths. Proceedings of SODA, pages 379–380, 2004.