Single-Source Stochastic Routing

Shuchi Chawla[*] Tim Roughgarden[†]

April 18, 2006

Abstract

We introduce and study the following model for routing uncertain demands through a network. We are given a capacitated multicommodity flow network with a single source and multiple sinks, and demands that have known values but unknown sizes. We assume that the sizes of demands are governed by independent distributions, and that we know only the means of these distributions and an upper bound on the maximum-possible size. Demands are irrevocably routed one-by-one, and the size of a demand is unveiled only after it is routed.

A *routing policy* is a function that selects an unrouted demand and a path for it, as a function of the residual capacity in the network. Our objective is to maximize the expected value of the demands successfully routed by our routing policy. We distinguish between *safe* routing policies, which never violate capacity constraints, and *unsafe* policies, which can attempt to route a demand on any path with strictly positive residual capacity.

We design safe routing policies that obtain expected value close to that of an optimal unsafe policy in planar graphs. Unlike most previous work on similar stochastic optimization problems, our routing policies are fundamentally adaptive. Our policies iteratively solve a sequence of linear programs to guide the selection of both demands and routes.

[*]Microsoft Research, Silicon Valley Campus, Mountain View, CA. This work was performed while the author was visiting the Department of Computer Science, Stanford University and supported by DARPA grant W911NF-05-1-0224. Email: `shuchi@cs.wisc.edu`.

[†]Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Supported in part by ONR grant N00014-04-1-0725, DARPA grant W911NF-05-1-0224, and an NSF CAREER Award. Email: `tim@cs.stanford.edu`.

1 Introduction

We introduce and study the following model for routing uncertain demands through a network. We are given a multicommodity flow network, defined by a directed graph $G = (V, E)$ with vertices V and edges E, a nonnegative capacity c_e on each edge $e \in E$, and a collection $(s_1, t_1), \ldots, (s_k, t_k)$ of source-sink pairs, also called *commodities*. Associated with each commodity i is a demand with a known nonnegative *value* v_i and an unknown size. Our goal is to choose routes for a subset of the demands to maximize the value of these demands without violating the edge capacities. In the special case of known demand sizes, this is the well known and difficult *unsplittable flow* problem.

Inspired by recent work of Dean, Goemans, and Vondrak [3, 4] on stochastic versions of the Knapsack and Set Packing problems, we adopt the following model for unknown demand sizes. We assume that the size of the ith demand is governed by a distribution with known mean μ_i, and that the sizes of different demands are independent. We also assume that there is known upper bound D_{max} on the maximum-possible size of a demand. No other information about the size distributions is available. We assume that commodities are routed one-by-one. When a commodity is selected, the size of its demand is unveiled only *after* it is routed. Decisions are irrevocable, and a previously routed demand cannot be removed from the network.

A *routing policy* is a function that selects an unrouted commodity (s_i, t_i) and an s_i-t_i path for it, as a function of the residual capacity in the network. While routing policies can be very complex, we will only be interested in routing policies defined by polynomial-time algorithms. A routing policy can be *adaptive*, in the sense that its decisions depend on the instantiated sizes of the previously routed commodities, or *non-adaptive*, in which case it simply specifies a fixed order in which the demands should be routed and fixed paths for routing them. There has been significant recent work proving upper and lower bounds on the *adaptivity gap*—the ratio between the objective function values of an optimal adaptive and non-adaptive policy, respectively—for various problems [3, 4, 5, 7]. The problems that we consider in this paper have a large (polynomial) adaptivity gap, even in networks of parallel links (see Example D.1). In contrast to previous work, which primarily studied non-adaptive policies for various problems, we focus on the design and analysis of near-optimal adaptive policies. Our objective is to maximize the expected value of the commodities successfully routed by our routing policy.

When demand sizes are stochastic, edge capacity constraints can be interpreted in several ways. The most stringent definition is to require that a routing policy respect every edge capacity with probability 1. We call a routing policy *safe* if it meets this definition and *unsafe* otherwise. When an unsafe routing policy routes a commodity in a way that violates some capacity constraints, we assume that no value is obtained for this unsuccessfully routed commodity, and that all violated edges drop out of the network.

Both safe and unsafe policies have their advantages. Unsafe policies are clearly more general than safe ones, and could conceivably obtain a much larger expected value. Safe policies guarantee successful transport for all admitted commodities; this property is clearly desirable, and could be essential in certain applications.

In this work, we seek the best of both worlds: we will design only safe routing policies, but will bound their performance relative to an optimal *unsafe* routing policy. This goal is somewhat analogous to previous work [3, 4] that designed non-adaptive policies with expected value close to that obtained by an optimal adaptive policy.

Pursuing this ambitious goal forces us to adopt an additional assumption. To motivate it, consider the following example. Fix a value $\alpha \in (0, 1]$, let $\epsilon > 0$ be much smaller than α, and let $\delta > 0$ be much smaller than ϵ. Consider a network with two vertices s, t and one directed edge (s, t) with unit capacity. Suppose there are a large number of commodities, each with source s, sink t, unit value, and with size equal to α with probability δ and to ϵ with probability $1 - \delta$. A safe routing policy must cease routing commodities after at most $(1 - \alpha)/\epsilon$ commodities have been routed. On the other hand, an unsafe policy will typically route roughly $1/\epsilon$ commodities successfully, provided δ is sufficiently small. Thus safe policies might only capture a $1-\alpha$ fraction of the expected value of an optimal unsafe policy, where α is the maximum-possible fraction of an edge that a demand can occupy. For this reason, we assume throughout this paper that the maximum-possible size D_{max} of a commodity is bounded above by an $\alpha < 1$ fraction of the minimum edge capacity c_{min}. Similar but weaker assumptions are often made in the classical unsplittable flow problem [6, 12, 13]. Our goal will be to design routing policies that have good (constant or logarithmic) approximation ratios for every fixed α less than 1.

Achieving this goal in general multicommodity networks would give, as a special case, a fundamental breakthrough for solving the disjoint paths problem with constant congestion in directed graphs. On the other hand, the single-source unsplittable flow problem (with known demands) admits constant-factor approximation algorithms [6, 12, 13]. These facts motivate our second crucial assumption: we assume that all commodities share a common source vertex s. We call the problem of designing a routing policy for such an instance *Single-Source Stochastic Routing (SSSR)*.

1.1 Our Results

We first define a general algorithmic and analytical approach for designing near-optimal safe, adaptive routing policies for SSSR instances. Our algorithm uses a linear program (LP) to guide the commodity and route selection at each stage, and re-solves this LP each time a new commodity is routed. Our analysis framework is based on tracking the successive expected changes in the optimal objective function value of an LP-based upper bound on the expected value of an optimal (unsafe) routing policy, as our algorithm routes and instantiates demands.

As noted above, most previous work on related stochastic optimization problems [3, 4, 5, 7] has concentrated primarily on the design and analysis of non-adaptive policies; few techniques for designing adaptive policies are currently known. We believe that our iterative LP rounding approach could form the basis of near-optimal adaptive policies for a range of stochastic optimization problems.

We then apply this framework to obtain polynomial-time, safe routing policies with expected value close to that of an optimal unsafe policy for SSSR problems in planar graphs. (More generally, we only require that the supporting subgraph of a natural fractional flow relaxation is planar.) For general planar graphs, we achieve an approximation factor of $O((\log W)/(1 - \alpha))$, where $\alpha < 1$ is a constant satisfying $D_{max} \leq \alpha c_{min}$, and W denotes the maximum ratio between the "expected per-unit value" v_i/μ_i of two different commodities. Recall from the above example that the dependence on $1/(1 - \alpha)$ is necessary for this type of guarantee, even in single-link networks.

We also obtain a superior approximation factor of $O(1/(1-\alpha))$ in the special case where all of the sinks lie on a common face. This special case includes all outerplanar networks and all single-source, single-sink planar networks.

Finally, we discuss the limitations of our analysis framework for the SSSR problem in non-planar graphs.

1.2 Related Work

Starting with the work of Dantzig [2] in 1955, stochastic optimization problems have been studied extensively in Operations Research, primarily in the framework of Stochastic Programming (see, for example, [18] and [1]). While there is a vast theory on solving stochastic linear programs, there is little known about optimally or approximately solving stochastic versions of NP-hard optimization problems and stochastic integer programs. Part of the reason for this poor understanding is that the optimal solution to a stochastic optimization problem such as SSSR can be a complex, exponential-size decision diagram. In particular, the optimal decision at every stage of the problem can depend on the instantiations of the random variables in the previous stages.

Owing to this complexity, much work has focused on a simpler class of stochastic optimization problems called the k-*stage recourse model*. In these problems, the algorithm is required to pick a solution satisfying some feasibility constraints, and can do so in k stages. The feasibility constraints are drawn from a known distribution. In each stage, the algorithm learns additional information about the constraints, but the cost of picking the solution increases with each successive stage. The goal is to design an algorithm that incurs the minimum-possible expected cost.

Several recent works by the theoretical CS community have studied the recourse model. Starting with the work of Ravi and Sinha [15] and Immorlica et al. [11], constant-factor approximation algorithms have been developed for the 2-stage stochastic versions of problems such as Steiner tree, network design, facility location, and vertex cover (see e.g. [8, 9, 10, 16] and the references therein). Some of this work has been extended to the k-stage versions of these problems [9, 17], albeit with approximation factors that depend linearly or even exponentially on the number of stages k.

The work that is most closely related to ours is that of Dean, Goemans and Vondrak [3, 4, 7]. Dean et al. study stochastic versions of several packing and covering problems such as Knapsack, that are similar in flavor to our stochastic routing problem. For example, the Stochastic Knapsack problem is essentially SSSR in a single-link network, whereas SSSR in a general graph is similar to an instance of the Stochastic Multi-dimensional Knapsack problem, with a unique dimension corresponding to each edge of the graph.

However, our focus on routing applications leads to several key differences between their work and ours. First, in the SSSR problem, a routing policy must select both the next commodity to route, as well as *how* to route it. There is no analogue of this combinatorial route selection issue in the packing and covering problems studied in [3, 4], which primarily involve only binary decision variables. Second, capacity constraints are enforced differently in the work of Dean et al. than in the present paper. In [3, 4], unsafe policies are allowed, but such a policy must terminate as soon as a single constraint is violated. In the SSSR problem, an unsafe routing policy can continue to route the remaining commodities on edges that have not yet dropped out of the network. We believe that this less restrictive notion of an unsafe policy is more suitable for routing applications. Third, we design safe routing policies, whereas Dean et al. design policies that are unsafe in the above restricted sense. Thus while our guarantees are in some sense stronger than those in [3, 4], we prove such guarantees only under an additional assumption ($D_{max} \leq \alpha c_{min}$ for some $\alpha < 1$) that is not needed in the work of Dean et al. Finally, as noted earlier, Dean et al. focus on obtaining tight bounds on the adaptivity gap, whereas we seek adaptive solutions that achieve an approximation factor far smaller than the adaptivity gap.

2 The Stochastic Routing Model

We consider a directed network $G = (V, E)$ with edge capacities $c : E \rightarrow \Re^+$. indexed by $i \in I$, each with a source-sink pair (s_i, t_i) will assume that all commodities share a common source s. The "size" or demand of a commodity i is given by the random variable D_i, drawn from a distribution with mean $\mu_i = \mathbf{E}[D_i]$. We assume that the sizes of different demands are independent. For every commodity i, let $w_i = v_i/\mu_i$ denote its "expected per-unit value". We assume that commodities are ordered such that $w_1 \geq w_2 \geq \cdots \geq w_k$.

Let $D_{\max}$ be the smallest value d such that $\mathbf{Pr}[D_i > d] = 0$ for all $i \in I$. We assume that D_{max} is known to the algorithm and that $D_{\max} < c_{\min}$, where $c_{\min} = \min_e c_e$ is the minimum edge capacity in the graph. Let $\alpha < 1$ denote the ratio between $D_{\max}$ and c_{min}. As shown by the example in the Introduction, our approximation guarantees necessarily depend on the value of α.

Let $\mathcal{P}_i$ denote the s_i-t_i paths of G. A routing policy successively picks a commodity i and a path $P_i \in \mathcal{P}_i$ for routing it. After the algorithm picks a commodity and its corresponding path, the demand D_i for that commodity gets instantiated to some value d_i. If d_i is at most the minimum residual capacity of the edges of P_i, then the commodity is admitted and the algorithm obtains the value v_i. The algorithm continues until no more commodities can be admitted. The goal of the algorithm is to maximize the expectation of its total accrued value. As described previously, a routing policy is safe if every commodity picked by it gets admitted with probability one.

3 Approximation Algorithms via Iterative Rounding

In this section we give a general algorithmic and analytic approach for approximating stochastic routing problems; we apply these ideas to SSSR problems in planar graphs in the next section. Subsection 3.1 introduces a linear programming bound on the expected value achievable by an optimal (unsafe) routing policy, Subsection 3.2 introduces the high-level idea of our algorithm, and Subsection 3.3 defines our analytic framework.

3.1 An LP Relaxation for the Optimal Routing Policy

We begin with a linear program that provides an upper bound on the expected value of an optimal (unsafe) routing policy for a given stochastic routing instance:

$$
\begin{array}{lll}
LP(I,u): \quad \max & \sum_{i\in I} w_i \sum_{e\in\delta^+(s_i)} f_e^{(i)} \text{ s.t.} & \\
& \sum_{i\in I} f_e^{(i)} \le u_e & \forall e \in E \\
& \sum_{e\in\delta^+(s_i)} f_e^{(i)} \le \mu_i & \forall i \in I \\
& \sum_{e\in\delta^-(v)} f_e^{(i)} = \sum_{e\in\delta^+(v)} f_e^{(i)} & \forall i \in I, v \in V \setminus \{s_i,t_i\} \\
& f_e^{(i)} \ge 0 & \forall i \in I, P \in \mathcal{P}_i.
\end{array}
$$

Recall that w_i denotes the ratio v_i/μ_i. Also, $\delta^+(v)$ and $\delta^-(v)$ denote the sets of edges directed out of and into the vertex v, respectively. Note that $LP(I,u)$ is simply a standard LP formulation of the maximum-value (w.r.t. "values" w) multicommodity flow subject to edge capacities u and per-commodity flow rate constraints μ.

Proposition 3.1. *The expected value obtained by an optimal routing policy for a stochastic routing instance with commodities I and edge capacities c is at most $LP(I,(1+\alpha)c)$, where $\alpha = D_{max}/c_{min}$.*

Proposition 3.1 is similar to a result by Dean, Goemans, and Vondrak [3] in the special case of a single-link network (Knapsack), although our proof (Appendix B) is somewhat different. Scaling, we also obtain the following corollary.

Corollary 3.2. *For every $\gamma \in (0,1]$, the expected value obtained by an optimal routing policy for a stochastic routing instance with commodities I and edge capacities c is at most $\frac{1}{\gamma} \cdot LP(I,\gamma(1+\alpha)c)$, where $\alpha = D_{max}/c_{min}$.*

3.2 An Iterative Rounding Algorithm

We next develop a safe, adaptive routing algorithm that iteratively uses linear programs of the form $LP(I,u)$ to guide both commodity and route selections. The high-level idea of the algorithm is to scale down the given

Input: A stochastic routing instance G, c, I.
Output: A commodity $i \in I$ and a path $P \in \mathcal{P}_i$ at every step.

1. Initialize J to I and $\hat{c}_e = (1-\alpha)c_e$ for every $e \in E$. Solve $LP(J, \hat{c})$, obtaining an optimal solution $\hat{f}$.
2. While $\hat{f}_e^{(i)} > 0$ for some commodity $i \in J$ and edge $e \in E$:
   (a) Pick $i \in J$ and $P \in \mathcal{P}_i$ such that $\hat{f}_e^{(i)} > 0$ for every $e \in P$, and route the commodity i on P.
   (b) Set $J := J \setminus \{i\}$.
   (c) Set $\hat{c}_e := \max\{0, \hat{c}_e - d_i\}$ for every edge $e \in P$, where d_i is the instantiated size of commodity i.
   (d) Re-solve $LP(J, \hat{c})$, obtaining a new optimal solution $\hat{f}$.

Figure 1: High-level description of the algorithm IR.

edge capacities (to ensure safeness), solve $LP(I, u)$, pick the fractionally routed commodity with largest ratio w_i and route it on one of its (fractional) flow paths, and repeat. This high-level algorithm is given in Figure 1.

Algorithm IR is well defined and safe no matter how we implement Step 2a. (See Appendix B for a proof.)

Lemma 3.3. *Algorithm* IR *is a safe routing policy.*

To obtain good approximation results, however, we need to choose the commodity i and the path $P \in \mathcal{P}_i$ in Step 2a carefully. One natural refinement of Algorithm IR is to always choose a commodity i in Step 2a with maximum-possible ratio w_i; we call this variant the GREEDY-IR algorithm. We defer the much more subtle issue of path selection to the next subsection.

3.3 Path Coverings

We now describe our criterion for path selection in Step 2a of Algorithm IR. To motivate the next definition, suppose that in the first stage we pick a commodity i and an s_i-t_i flow path P. The size of commodity i might get instantiated to some value much larger than μ_i, which in turn could evict other commodities in the LP solution from the edges of P. Intuitively, our goal will be to pick a path to minimize the severity of this eviction. We make this idea precise with the following notion of r-*coverable* paths.

Definition 3.4. Fix a stochastic routing instance. Let $\{\hat{f}_e^{(i)}\}_{i,e}$ be a feasible solution to $LP(I, u)$. Let $\{\hat{f}_P^{(i)}\}_{i,P\in\mathcal{S}}$ be a flow decomposition of f, where $\mathcal{S} \subseteq \cup_i \mathcal{P}_i$ denotes the set of paths that carry a positive amount of flow.

(a) Let $P^* \in \mathcal{S}$ be a path with $f_{P^*}^{(i)} > 0$ and $\mathcal{S}' \subseteq \mathcal{S}$ a collection of flow paths for commodities other than i. Let $F^* \subseteq P^*$ denote the edges of P^* contained in some path of $\mathcal{S}'$. The set $\mathcal{S}'$ r-*covers* P^* if there are $q \leq r$ paths $P_1, \ldots, P_q \in \mathcal{S}'$ such that every edge of F^* lies in at least one path P_i.

(b) The path decomposition $\{\hat{f}_P^{(i)}\}$ r-*covers* the path $P^* \in \mathcal{S}$ if for every subset $\mathcal{S}' \subseteq \mathcal{S}$ of flow paths for commodities other than i, $\mathcal{S}'$ r-covers P^*.

(c) An s_i-t_i path P^* with $\hat{f}_e^{(i)} > 0$ for every $e \in P^*$ is r-*coverable* if there exists a path decomposition with $\hat{f}_{P*}^{(i)} > 0$ that r-covers P^*.

Intuitively, increasing the amount of flow on an r-coverable path only evicts flow from r other flow paths.

Example 3.5. In a stochastic routing instance in a single-link network (i.e.., Knapsack), every flow path is 1-coverable. More generally, a flow path that contains at most ℓ edges is ℓ-coverable.

We next prove the central result of this section: if Algorithm GREEDY-IR can be implemented to route commodities only on r-coverable paths, then its expected value is at least an $\Omega(1/r)$ fraction of the expected value of an optimal (unsafe) routing policy.

Lemma 3.6. *If Algorithm* GREEDY-IR *routes commodities only on r-coverable paths, then its expected value is at least a $(1-\alpha)/(r+1)(1+\alpha)$ fraction of that of an optimal routing policy.*

Proof. Fix an execution of Algorithm GREEDY-IR on a stochastic routing instance. Let h denote the number of times that the main while loop executes. Relabel the commodities $I = \{1, \ldots, k\}$ so that the ith commodity is routed in iteration i. Set $I^0 = I$ and I^j equal to $\{j+1, \ldots, k\}$, the commodities remaining after the first $j \leq h$ iterations. Set $c^0 = (1-\alpha)c$ and let c^j denote the residual capacities $\hat{c}$ after the first j commodities have been routed. By the stopping condition, $LP(I^h, c^h) = 0$.

Our key claim is that for every $j \in \{1, 2, \ldots, h\}$,

$$LP(I^{j-1}, c^{j-1}) - LP(I^j, c^j) \leq r \cdot w_j \cdot d_j + v_j, \tag{1}$$

where d_j is the instantiated size of commodity j. Conceptually, this claim asserts that each time we route a new commodity, the amount by which the value of $LP(I^j, c^j)$ decreases is not much more than the additional value that we accrue. Since the initial value $LP(I, c^0)$ is comparable to the expected value of an optimal routing policy (by Corollary 3.2), this ensures that, in expectation, Algorithm GREEDY-IR will capture a significant (roughly $1/r$) fraction of the maximum-possible expected value.

To prove the claim, fix j and let P^* denote the path on which Algorithm GREEDY-IR routes commodity j. By assumption and the definition of r-coverable, there is a flow decomposition $\{\hat{f}_P^{(i)}\}$ of an optimal solution $\hat{f}$ to $LP(I^{j-1}, c^{j-1})$ that r-covers P^*. Let $\mathcal{S}$ denote the paths that carry a positive amount of flow in this decomposition. We next massage this path decomposition into a feasible solution for $LP(I^j, c^j)$ in two steps. For an edge $e \in P^*$, let $\hat{f}_e^{(-j)}$ denote the flow $\sum_{i \neq j} \hat{f}_e^{(i)}$ on edge e belonging to commodities other than j. We first decrease flow on paths of $\mathcal{S}$ for commodities other than j until the the flow of every edge $e \in P^*$ has decreased by at least $\min\{f_e^{(-j)}, d_j\}$. We then remove all flow paths corresponding to commodity j. Since $c_e^j = \max\{0, c_e^{j-1} - d_j\}$ for $e \in P^*$ and $c_e^j = c_e^{j-1}$ for $e \notin P^*$, these two steps will define a flow g feasible for $LP(I^j, c^j)$.

We now elaborate on the first step. Initialize $g_P^{(i)}$ to $f_P^{(i)}$ for all paths $P \in \mathcal{S}$. Let $F^* \subseteq P^*$ denote the edges of P^* from which flow still needs to be removed, in the sense that $f_e^{(-j)} - g_e^{(-j)} < \min\{f_e^{(-j)}, d_j\}$. While $F^* \neq \emptyset$, we decrease flow on paths of $\mathcal{S}$ as follows. Consider the paths P of $\mathcal{S}$ with $g_P^{(i)} > 0$ and $P \cap F^* \neq \emptyset$. Each edge of F^* lies in at least one such path. Since the original flow decomposition of $\hat{f}$ r-covers P^*, there are $q \leq r$ such paths $P_1, \ldots, P_q$ that collectively contain all of the edges of F^*. We decrease the corresponding value of $g_P^{(i)}$ for each of these paths at a uniform rate, until either $f_e^{(-j)} - g_e^{(-j)} = \min\{f_e^{(-j)}, d_j\}$ for some edge $e \in F^*$, or until $g_P^{(i)}$ is decreased to 0 for one of the paths $P_1, \ldots, P_q$. We denote by Δ_ℓ the amount by which the flow on $P_1, \ldots, P_q$ is decreased during the ℓth iteration of this procedure.

As long as $F^* \neq \emptyset$, we can perform the above operation to decrease flow. Every iteration strictly decreases the sum of $|F^*|$ and the number of paths of $\mathcal{S}$ that carry flow in g. The above procedure must therefore terminate with a final flow g. After deleting all of the flow paths corresponding to the commodity j, the flow g is feasible for $LP(I^j, c^j)$.

We complete the proof of the key claim by comparing the objective function values of $\hat{f}$ and g. First, we have

$$w_j \sum_{P \in \mathcal{P}_j} \hat{f}_P^{(j)} \leq w_j \cdot \mu_j = v_j. \tag{2}$$

Second, consider the flow decrease operations used to obtain the final flow g from $\hat{f}$. Every such operation decreases flow on at most r paths. Also, since every such operation decreases the amount of flow on every edge of F^*, the total flow decrease $\sum_{\ell \geq 1} \Delta_\ell$ over all such operations is at most d_j. Thus

$$\sum_{i \in I_j} \sum_{P \in \mathcal{P}_i} \left(\hat{f}_P^{(i)} - g_P^{(i)} \right) \leq r \cdot d_j.$$

By the definition of Algorithm GREEDY-IR, $w_j \geq w_i$ for every commodity $i \in I^j$ with $\hat{f}_e^{(i)} > 0$ for some $e \in E$. Hence

$$\sum_{i \in I^j} w_i \sum_{P \in \mathcal{P}_i} \hat{f}_P^{(i)} - \sum_{i \in I^j} w_i \sum_{P \in \mathcal{P}_i} g_P^{(i)} \leq r \cdot d_j \cdot w_j. \tag{3}$$

Since $\hat{f}$ and g are optimal and feasible solutions to $LP(I^{j-1}, c^{j-1})$ and $LP(I^j, c^j)$, respectively, adding the inequalities (2) and (3) proves the claim (1).

With the key claim in hand, we now complete the proof of the lemma. First, for a fixed execution of Algorithm GREEDY-IR, we can sum (1) over all $j \in \{1, 2, \ldots, h\}$ to obtain

$$\frac{1-\alpha}{1+\alpha} \cdot OPT \leq LP(I, (1-\alpha)c) \leq \sum_{i \in I^h} v_i \left(r \frac{d_i}{\mu_i} + 1 \right), \tag{4}$$

where the first inequality follows from Corollary 3.2 with $\gamma = (1-\alpha)/(1+\alpha)$, and in the second inequality we are using the equalities $w_i = v_i/\mu_i$ and $LP(I^h, c^h) = 0$.

Finally, consider a random execution of the algorithm GREEDY-IR. Label the commodities $1, 2, \ldots, k$ in an arbitrary way. Let X_i denote the indicator variable for the event that Algorithm GREEDY-IR attempts to route commodity i, and D_i the random variable equal to the size of commodity i. As in the proof of Proposition 3.1, the random variables X_i and D_i are independent for each i. Taking expectations in (4), we have

$$\begin{aligned} \frac{1-\alpha}{1+\alpha} \cdot OPT &\leq \mathbf{E}\left[\sum_{i=1}^{k} X_i \cdot v_i \left(r \frac{D_i}{\mu_i} + 1 \right) \right] \\ &= r \sum_{i=1}^{k} \frac{v_i}{\mu_i} \mathbf{E}[X_i \cdot D_i] + \sum_{i=1}^{k} v_i \mathbf{E}[X_i] \\ &= r \sum_{i=1}^{k} \frac{v_i}{\mu_i} \mathbf{E}[X_i] \cdot \mathbf{E}[D_i] + \sum_{i=1}^{k} v_i \mathbf{E}[X_i] \qquad (5) \\ &= (r+1) \sum_{i=1}^{k} v_i \mathbf{E}[X_i], \qquad (6) \end{aligned}$$

where (5) follows from the independence of X_i and D_i. Since Algorithm GREEDY-IR is a safe routing policy (Lemma 3.3), the sum on the right-hand side of (6) is precisely the expected value obtained by Algorithm GREEDY-IR, and the proof is complete. □

To usefully apply Lemma 3.6, there must be a commodity i that meets two orthogonal criteria: a large ratio w_i and a flow path that is r-coverable for small r. When the maximum variation w_1/w_k in expected per-unit values is small, the choice of commodity can be dictated by the second criterion alone. Precisely, we have the following variation on Lemma 3.6, which will be useful in Section 4.

Lemma 3.7. *If Algorithm* IR *routes commodities only on r-coverable paths, then its expected value is at least a $(1-\alpha)/(rW+1)(1+\alpha)$ fraction of that of an optimal routing policy, where $W = w_1/w_k$.*

Proof. (Sketch) Follow the proof of Lemma 3.6, replacing the right-hand side of inequality (3) by $r \cdot d_j \cdot w_j \cdot W$. □

4 Iterative Rounding in Planar Graphs

We now consider the SSSR problem in planar graphs and show the existence of r-coverable paths in them. In particular, we show that there always exists a 2-coverable commodity in a planar flow and give an algorithm for finding it (Section 4.1). Unfortunately, this is not necessarily the commodity with the maximum per-unit value w_i. (See Appendix A for an example of a planar SSSR instance where the maximum per-unit value commodity is only $\Theta(\log k)$-coverable). However, limiting our solution to a subset of commodities that have comparable w_i values, we obtain an $O(\log W)$ approximation for general planar graphs, where $W = w_1/w_k$ (Section 4.3).

We obtain a constant-factor approximation in the special case where all of the sinks lie on a common face in some embedding of the planar network. Here, we show that every commodity, and in particular the one with the maximum per-unit value, has a 2-coverable path (Section 4.2). Lemma 3.6 then implies that the GREEDY-IR algorithm achieves a constant-factor approximation for such instances.

4.1 Preliminaries

Let $G = (V, E)$ be a planar multicommodity flow network with a single source s, and f a feasible flow. Let $g : V \rightarrow \Re^2$ be a straight-line planar embedding of G. Such an embedding always exists [19].

A non-crossing path-decomposition

Recall that $\{f_P^{(i)}\}_{P \in \mathcal{S}}$ denotes a path-decomposition of f with $\mathcal{S}$ being the set of flow-carrying paths. We are interested in path decompositions of planar flows that are *non-crossing*, as defined below.

Definition 4.1. A path P *crosses* another path P' if there exists a bounded connected region X in $\Re^2$ with the following properties: P and P' each cross the boundary of X exactly twice and these crossings are interleaved. Precisely, if we scan the boundary of X in clockwise direction starting from the point where P enters it, we encounter P' exactly once before we see P again (see Figure 3(a) in the appendix). The set X is called a *witness* to this crossing of P and P'.

Definition 4.2. A set of paths is said to be *non-crossing* if every pair of paths is distinct and non-crossing.

Non-crossing path decompositions of single-source planar flows always exist and can be found in polynomial time. Appendix C contains a proof of this fact.

Lemma 4.3. *Every single-source planar multicommodity flow f has a non-crossing path decomposition that can be found in polynomial time.*

Given a non-crossing path-decomposition $\{f_P\}_{P \in \mathcal{S}}$, we can pick a small cover for a path as follows. We order all the paths in anticlockwise order. (This is well defined because no two paths cross.) Then for any path, roughly speaking, the two paths immediately neighboring the path should cover all its intersections with other paths.

More formally, we define a linear order $\prec$ on paths as follows. We order all the edges incident on s in anticlockwise order, starting from an arbitrary edge. This divides the paths $P \in \mathcal{S}$ into groups $\mathcal{S}_e$ based on the first edge in each path. If the edge e_1 precedes edge e_2 in anticlockwise order, then for all $P_1 \in \mathcal{S}_{e_1}$ and $P_2 \in \mathcal{S}_{e_2}$, we have $P_1 \prec P_2$. We then refine the ordering in each group. For group $\mathcal{S}_e$ with $e = u \to v$, consider all edges outgoing from v, and order them in anticlockwise order starting from e. This subdivides the group $\mathcal{S}_e$ into subgroups $\mathcal{S}_{e'}$ based on the next edge e' in each path. As before, if the edge e'_1 precedes edge e'_2 in anticlockwise order, then for all $P_1 \in \mathcal{S}_{e'_1}$ and $P_2 \in \mathcal{S}_{e'_2}$, we have $P_1 \prec P_2$. We continue in this manner until we obtain a total order. We rename the paths according to this order so that $P_1 \prec \cdots \prec P_q$ with $q = |\mathcal{S}|$.

Undominated commodities

Fix a non-crossing flow decomposition of a planar single-source multicommodity flow and a flow path P. Above, we suggested covering a path P using the two immediately neighboring paths. This is not sufficient to cover all of the intersection between P and other flow paths if, informally, the neighboring paths are "shorter" than P. To dodge this issue, we define a partial order on the commodities, roughly in order of the source-sink distance, and pick the commodity that is the "closest" to the source according to this order.

For a commodity i, let E_i denote the set of edges from which t_i is reachable along flow-carrying edges. Let $\mathcal{A}_i$ denote the subset of $\Re^2$ enclosed by this set of edges (including the edges, but not the embedding $g(t_i)$ of t_i). We call this set the *region enclosed by* i.

Definition 4.4. A commodity i *dominates* a commodity j if $g(t_i) \in \mathcal{A}_j$.

The dominance relation defines a partial order on commodities. The following lemma is proved in Appendix C.

Lemma 4.5. *If i dominates j, then $\mathcal{A}_i \subset \mathcal{A}_j$.*

Corollary 4.6. *The dominance relation is antisymmetric; hence, there exists an undominated commodity.*

4.2 Undominated Commodities are 2-coverable

We now show that for every planar single-source multicommodity flow, there is at least one 2-coverable flow path.

Lemma 4.7. *Let $\{f_P^{(i)}\}_{P \in \mathcal{S}}$ be a non-crossing path decomposition of the planar, single-source multicommodity flow flow f. Let i be an undominated commodity. Then every commodity i flow path in $\mathcal{S}$ is 2-covered by the path decomposition $\{f_P^{(i)}\}_{P \in \mathcal{S}}$.*

Proof. Let $P_1 \prec \cdots \prec P_q$ be a linear order on $\mathcal{S}$ as defined in the previous subsection. Consider a commodity i flow path $P = P_l \in \mathcal{S}$ and let $\mathcal{S}' \subseteq \mathcal{S}$. Let $x_1 = \text{argmax}_{x<l}\{P_{x \bmod q} \in \mathcal{S}'\}$ and $x_2 = \text{argmin}_{x>l}\{P_{x \bmod q} \in \mathcal{S}'\}$. Let $Q_1 = P_{x_1 \bmod q}$ and $Q_2 = P_{x_2 \bmod q}$. We claim that $\{Q_1, Q_2\}$ covers P with respect to $\mathcal{S}'$.

Suppose the claim does not hold. Then there exists a path $R \in \mathcal{S}'$ with $R \cap P \not\subseteq Q_1 \cup Q_2$. Assume that $R \prec Q_1 \prec P$; the other cases can be handled analogously. Let v be the first vertex on P that is in R but not in Q_1. Consider the region X enclosed by the segments of P and R between s and v. Then it is immediate that $X \subseteq \mathcal{A}_i$ because t_i is reachable from v, and therefore also from any edge on the segments of P and R between s and v.

Furthermore, recall that $R \prec Q_1 \prec P$. Therefore, some prefix of Q_1 lies within the region X. However, since Q_1 does not cross P or R, it can only exit X from the vertex v. Then $v \notin Q_1$ implies that Q_1 lies entirely within the region X, and the terminal corresponding to Q_1, say t_j, lies within $\mathcal{A}_i$. However, this implies that j dominates i and contradicts the fact that i is undominated. □

Lemmas 3.7 and 4.7 easily imply a constant-factor approximation ratio for the GREEDY-IR algorithm when all sinks like on a common face in some planar embedding.

Theorem 4.8. *In a planar instance of SSSR in which all sinks lie on a single face, algorithm* GREEDY-IR *achieves a* $\left(3\frac{(1+\alpha)}{(1-\alpha)}\right)$*-approximation.*

Proof. Let f be a single-source flow multicommodity flow in a planar graph G, where all of the sinks lie on a single face in some embedding of G. We first transform the graph such that each sink is a leaf. We accomplish this by adding, for every sink t_i, a new vertex t'_i and an infinite capacity edge from t_i to t'_i. Let g be a straight line embedding of this augmented graph with each t'_i lying on the outer face. The flow f extends to this augmented graph in an obvious way. We claim that every commodity of f is undominated. In particular, note that for every pair of distinct commodities i and j, i is not reachable from the edge $t_j \to t'_j$. Furthermore, by definition, $\mathcal{A}_i$ does not contain the outer face of the embedding g. Therefore, $t'_j \notin \mathcal{A}_i$. Since all commodities are undominated, Lemmas 3.7 and 4.7 imply the result. □

Of course, Theorem 4.8 includes the special cases of outerplanar networks and of single-source, single-sink planar instances of SSSR.

4.3 An $O(\log W)$-Approximation for General Planar Graphs

In the previous subsection we showed that there always exists a 2-coverable commodity in a planar flow. Unfortunately, as Example A.1 in Appendix A shows, the most-profitable commodity (with the highest w_1) may not be $o(\log k)$-coverable. However, as we show below, having at least one 2-coverable commodity in any planar graph instance is sufficient to obtain an $O(\log W)$-approximation, where $W = w_1/w_k$ is the ratio between the maximum and minimum per-unit values.

We can assume via scaling that the minimum per-unit value w_k is 1. We divide the commodities into $\log W$ groups:

$$I_x = \left\{i : w_i \in [2^x, 2^{x+1})\right\} \quad x \in \{0, \cdots, \log W\}.$$

Algorithm PLANAR-IR proceeds as follows. We consider the optimal values $\mathcal{V}_x$ of $\log W$ linear programs $LP(I_x, (1-\alpha)c)$, one for each group I_x. These values give us an estimate of the total value that an optimal adaptive solution can derive from each group of commodities. Let x^* be the index of the group for which the maximum value $\mathcal{V}_x$ is achieved. We run the algorithm IR on the graph using only commodities in the group I_{x^*}. (In other words, we round the flow obtained by solving the $LP(I_{x^*}, (1-\alpha)c)$.) In step 2a of the algorithm, we pick an undominated commodity and route it along a flow path in a non-crossing path decomposition of the flow.

Lemma 4.9. *Algorithm* PLANAR-IR *is a* $(5\frac{1+\alpha}{1-\alpha}\log W)$*-approximation.*

Proof. Consider the optimal solution f to $LP(I,(1-\alpha)c)$, and let f_x denote the flow due to commodities in group I_x in this solution. Let $\mathcal{V}_x^f$ denote the value contributed to the objective function by the group I_x and let $\mathcal{V}^f$ denote the total value $\sum_x \mathcal{V}_x^f$. Note that f_x is a feasible solution to $LP(I_x,(1-\alpha)c)$, and therefore, $\mathcal{V}_x^f \le \mathcal{V}_x$. Furthermore, $\max_x V_x^f \ge \frac{1}{\log W}\mathcal{V}^f$. Therefore, $\mathcal{V}_{x^*} \ge \frac{1}{\log W}\mathcal{V}^f$.

Now Lemma 4.7 implies that we always route a commodity along a 2-coverable path in step 2a of the algorithm PLANAR-IR. Furthermore, the per-unit value of the commodity routed in each step is at least half the per-unit value of any other commodity in the set I_{x^*}. Lemma 3.7 then implies that the expected value obtained by the PLANAR-IR algorithm is at least a $1/5$ fraction of $\mathcal{V}_{x^*}$, and is thus at least a $\frac{1-\alpha}{5(1+\alpha)}\frac{1}{\log W}$ fraction of the expected value obtain by an optimal routing policy for all of the demands. □

5 Conclusions and Open Problems

We developed a general technique for approximating the single-source stochastic routing problem in planar graphs via adaptive policies. We conclude with some promising directions for future work.

First, can a polynomial-time routing policy obtain a non-trivial constant or logarithmic approximation of optimal in non-planar graphs? Example A.2 shows that we cannot prove an $O(\log k)$-approximation, even in single-sink instances, using our notion of coverability. However, this example does not preclude better approximation guarantees via a different analysis—in fact, algorithm GREEDY-IR achieves a constant-factor approximation in this example. The limitation of our technique is in bounding the decrease in the LP's value in *each* step of the process. Can we do better by bounding the total loss in the LP's value over all steps? Second, the iterative rounding approach seems general enough to be applicable to other stochastic optimization problems. In the case of routing, the notion of coverability allows us to track the change in the optimal value of each successive LP. Are there similar notions for other optimization problems? Third, our LP upper bound (Proposition 3.1) is weak in that it ignores the order in which the optimal strategy schedules commodities. Is there a better (but analytically tractable) upper bound that makes use of this ordering information?

References

[1] John R. Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Series in Operations Research. Springer-Verlag, New York, 1997.

[2] George B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[3] B. Dean, M. Goemans, and J. Vondrak. The benefit of adaptivity: Approximating the stochastic knapsack problem. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 208–217, 2004.

[4] B. Dean, M. Goemans, and J. Vondrak. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 395–404, 2005.

[5] Brian Dean. *Approximation Algorithms for Stochastic Scheduling Problems*. PhD thesis, Massachusetts Institute of Technology, Massachusetts, 2005.

[6] Yefim Dinitz, Naveen Garg, and Michel X. Goemans. On the single-source unsplittable flow problem. *Combinatorica*, 19(1):17–42, 1999.

[7] M. Goemans and J. Vondrak. Stochastic covering and adaptivity. In *LATIN*, pages 532–543, 2006.

[8] A. Gupta, M. Pal, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 417–426, 2004.

[9] A. Gupta, M. Pal, R. Ravi, and A. Sinha. What about wednesday? approximation algorithms for multistage stochastic optimization. In *Proceedings of the8thInternational Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, 2005.

[10] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In *Proceedings of the45thAnnual IEEE Symposium on Foundations of Computer Science*, pages 218–227, 2004.

[11] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 684–693, 2004.

[12] Jon Kleinberg. Single-source unsplittable flow. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–77, 1996.

[13] S. G. Kolliopoulos and C. Stein. Approximation algorithms for single-source unsplittable flow. *SIAM Journal on Computing*, 31(3):919–946, 2001.

[14] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK, 1995.

[15] R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. *Mathematical Programming*, 2005.

[16] David Shmoys and Chaitanya Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Proceedings of the45thAnnual IEEE Symposium on Foundations of Computer Science*, pages 228–237, 2004.

[17] David Shmoys and Chaitanya Swamy. Sampling-based approximation algorithms for multi-stage stochastic optimization. In *Proceedings of the46thAnnual IEEE Symposium on Foundations of Computer Science*, 2005.

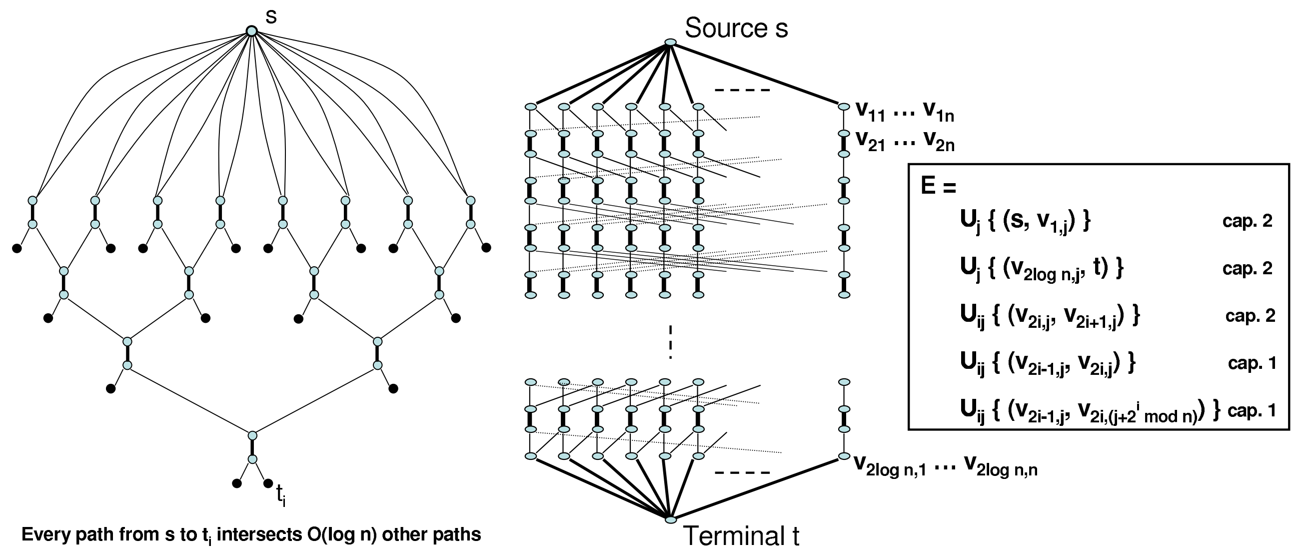
[18] Stochastic programming community homepage. http://stoprog.org/.

[19] W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3(13):743–768, 1963.

A Limitations of the Algorithm IR

In this section we present some examples exhibiting the limitation of algorithm IR in obtaining a good approximation for SSSR in general planar graphs and non-planar graphs.

We first show that in a multiple-sink planar graph instance, the maximum per-unit value commodity may not be $o(\log k)$-coverable.



(a) (b)

Figure 2: (a) A planar graph instance in which no flow paths for the maximum per-unit value commodity are r-coverable with $r = o(\log k)$; (b) A non-planar graph instance in which no source-sink paths are r-coverable with $r = o(\log k)$

Example A.1. Consider the instance in Figure 2(a). There are k commodities, all but one having value $v_j = 1/2$ and mean size $\mu_j = 1$. The filled circles in the figure depict the terminals for these commodities. The exception is the commodity i shown in the figure that has value $v_i = 1$ and $\mu_i = 1$. The thick edges in the graph have capacity 2 and all other edges have capacity 1. Then, in the optimal LP solution, each edge in the graph is saturated. It is also evident from the figure that any flow path for commodity i encounters $\log k$ other flow paths along its way, corresponding to the terminals along that path. Therefore no flow path for commodity i is r-coverable with $r < \log k$.

In the above example, although the most profitable commodity does not have a path with a small cover, there exist other commodities that are 2-coverable. Next we give an example of a non-planar single-source single-sink instance in which *no* commodity is r-coverable with $r = o(\log k)$.

Example A.2. Consider the graph in Figure 2(b). There are $k = 2n$ single-source single-sink commodities, with $\mu_i = 1$ for all i. The graph contains $2n \log n + 2$ nodes. Assume that n is a power of 2.

Consider any flow path P. Note that it carries at most one unit of flow, because the minimum capacity along any path is at most one. We claim that this path has a cover of size at least $\log n - 1$ in *any* path-decomposition of the flow. Note that all edges in the graph are saturated. Since P contains at least $\log n + 1$ edges of capacity 2, it intersects other paths along these capacity 2 edges. We claim that no path can intersect P along two edges of the form $(v_{2i,j}, v_{2i+1,j})$ (intermediate capacity 2 edges). This immediately implies that P has a cover of size at least $\log n - 1$, and we are done.

We now prove the claim. Suppose P' intersects P at two intermediate capacity 2 edges, and let $e = (v_{2i,j}, v_{2i+1,j})$ be the first of these. Then the next edges following e on the two paths must be different. One of these is $(v_{2i+1,j}, v_{2i+2,j})$ and the other $(v_{2i+1,j}, v_{2i+2,j+2^{i+1} \bmod n})$. Let us track how the second subscript of nodes in the two paths changes in the subsequent levels. At each level, the second subscript either stays the same or increases by a power of 2 larger than $i+1$ (modulo n, which is also a larger power of 2). Since the two start out being a distance of 2^{i+1} apart, they only converge in the end at vertex t. Therefore, the claim holds.

B The Linear Program Bound and Algorithm IR: Proofs for Section 3

Proof of Proposition 3.1. Fix a stochastic routing instance and an optimal routing policy for it. For a commodity i and an s_i-t_i path P, let X_P^i denote the indicator variable for the policy attempting to route commodity i on the path P. As usual, S_i denotes the (random) size of commodity i.

The successfully routed commodities must obey edge capacity constraints, and $S_i \leq \alpha c_{min}$ for all i with probability 1. Therefore, with probability 1 we have

$$\sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i : e \in P} X_P^i \cdot S_i \leq c_e + \alpha c_{min} \leq (1+\alpha) c_e \tag{7}$$

for every edge e. By the Principle of Deferred Decisions [14], the random variables X_P^i and S_i are independent for all i and P. Taking expectations in (7) then yields

$$\sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i : e \in P} \mathbf{Pr}\big[X_P^i = 1\big] \cdot \mu_i \leq (1+\alpha) c_e$$

for all edges e. For all $i \in I$, set $f_P^{(i)} = \mathbf{Pr}\big[X_P^i = 1\big] \cdot \mu_i$ for all $P \in \mathcal{P}_i$ and $f_e^{(i)} = \sum_{P \in \mathcal{P}_i : e \in P} f_P^{(i)}$ for all $e \in E$. Then, $\{f_e^{(i)}\}_{i,e}$ is a feasible solution to $LP(I, (1+\alpha)c)$. The objective function value of this solution of this solution is

$$\sum_{i \in I} \sum_{P \in \mathcal{P}_i} w_i \cdot \mathbf{Pr}\big[X_P^i = 1\big] \mu_i = \sum_{i \in I} \sum_{P \in \mathcal{P}_i} v_i \cdot \mathbf{Pr}\big[X_P^i = 1\big].$$

Since the expected value obtained by the optimal policy is only

$$\sum_{i \in I} \sum_{P \in \mathcal{P}_i} v_i \cdot \mathbf{Pr}[i \text{ successfully routed on } P],$$

the value $LP(I, (1+\alpha)c)$ is at least the expected value obtained by the optimal routing policy. □

Proof of Lemma 3.3. If algorithm IR attempts to route commodity i on the path P, then $\hat{c}_e > 0$ for every $e \in P$. Since $\hat{c}_e = (1-\alpha)c_e$ for every $e \in E$ initially, the total amount of flow routed on each edge e of P in previous iterations is at most $(1-\alpha)c_e$. Since the size of commodity i is at most αc_{min} with probability 1, every edge of P will carry at most c_e units of flow after commodity i is (successfully) routed. □

C Non-Crossing Paths and the Dominance Relation: Proofs for Section 4

Proof of Lemma 4.3. We start with an arbitrary path-decomposition $\{f_P^{(i)}\}_{P \in \mathcal{S}}$ of f and modify it in steps, decreasing the number of pairs of crossing paths by one at every step. We assume without loss of generality that every flow path carries the same amount of flow, that is, $f_P^{(i)} \in \{0, \varepsilon\}$ for all $i \in I$, $P \in \mathcal{S}$, and some fixed value ε. (This can be ensured easily by splitting each path into multiple paths carrying smaller

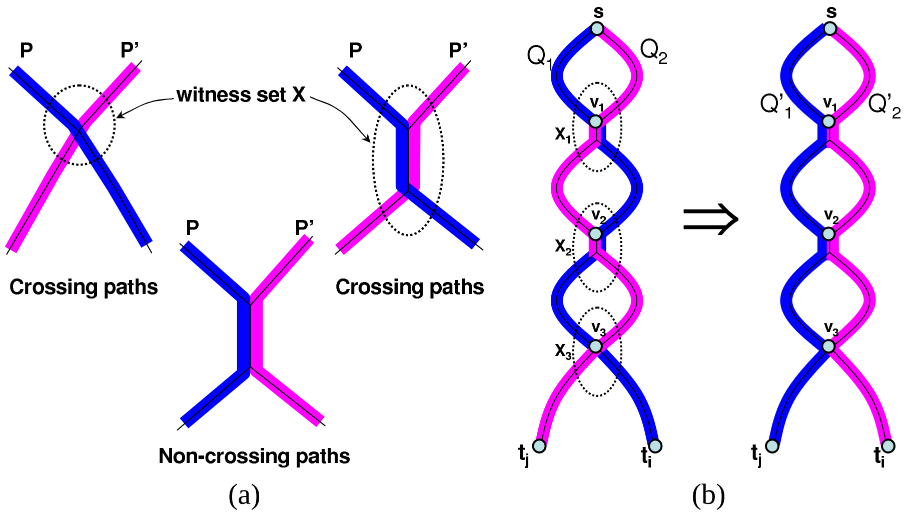


Figure 3: (a) Crossing and non-crossing paths; (b) Converting a crossing path-decomposition to a non-crossing one (see Lemma 4.3)

amounts of flow. If all the edge capacities are integral and poly-bounded, then the number of flow paths only increases by a polynomial factor.)

In each step, we pick a pair of crossing flow paths. Let these be Q_1, a commodity i flow path, and Q_2, a commodity j flow path (see Figure 3(b)). Note that the paths may cross multiple times. Let the regions $X_1, X_2, \cdots, X_q$ be the minimal witnesses to the crossings of these paths. For each $l \in [q]$ let v_l be the first vertex in X_l that lies in both Q_1 and Q_2. For $0 \leq l \leq q$ define Q_1^l to be the segment of Q_1 from v_l to v_{l+1}, where v_0 is defined to be s and v_{q+1} is defined to be t_i. The segments Q_2^l are defined analogously.

Let Q_1' be the path obtained by taking the union of segments Q_1^l with even l and Q_2^l with odd l. Likewise, let Q_2' be the path obtained by taking the union of segments Q_1^l with odd l and Q_2^l with even l. Then, assuming without loss of generality that q is odd (as in the figure), we have $Q_1' \in \mathcal{P}_j$ and $Q_2' \in \mathcal{P}_i$.

We modify the flow decomposition as follows: let $\mathcal{S} = \mathcal{S} \setminus \{Q_1, Q_2\} \cup \{Q_1', Q_2'\}$, $f_{Q_1}^{(i)} = f_{Q_2}^{(j)} = 0$, and, $f_{Q_1'}^{(j)} = f_{Q_2'}^{(i)} = \varepsilon$. It is immediate from construction that the paths Q_1' and Q_2' do not cross. Furthermore, if there is a flow path R that crosses only one of Q_1 and Q_2, then it crosses only one of Q_1' and Q_2'. Therefore, the number of crossing path pairs in the new decomposition is one less than the number of crossing pairs in the previous decomposition.

We repeat the above procedure until no crossing pairs remain. The algorithm ends after at most $|\mathcal{S}|^2$ steps and we obtain a non-crossing path-decomposition. □

Proof of Lemma 4.5. First note that $\mathcal{A}_i \neq \mathcal{A}_j$, because $g(t_i)$ lies in $\mathcal{A}_j$ but not in $\mathcal{A}_i$. Suppose that the claim does not hold for some pair of commodities i and j. Then there exists an edge e such that $e \in \mathcal{A}_i \cap E_i$ but $e \notin \mathcal{A}_j$. Then, by definition, there exists a directed path to t_i containing e. Since $g(t_i) \in \mathcal{A}_j$, the path must cross the boundary of $\mathcal{A}_j$. Let v be a vertex where the path crosses the boundary. Then there is a directed path from v to t_j (since v lies on the boundary of $\mathcal{A}_j$). But this implies that there exists a path from e to t_j via v, contradicting the fact that $e \notin \mathcal{A}_j$. □

D Non-Adaptive Routing Policies

In this section we present nearly matching upper and lower bounds on the adaptivity gap of stochastic routing. We start with an example establishing a lower bound on the adaptivity gap. This is similar to the example described in the Lemma 2.1 of [3]. We present a proof sketch of the lower bound.

Example D.1. Consider a single-source single-sink network with n parallel edges of capacity 1 each between the source and sink. There are $k = n^2$ commodities with demand 1 w.p. $p = 1/\sqrt{n}$ and 0 w.p. $1-p$. Each commodity has value 1. The adaptive optimal solution gets $n\sqrt{n}$ commodities in expectation, and has value $n\sqrt{n}$. On the other hand, an easy (rough) calculation shows that any non-adaptive solution gets at most $O(n)$ commodities in expectation, getting a gap of $\Omega(\sqrt{n})$. To see this, note that the best non-adaptive strategy is load balancing or round robin—route the ith commodity via the (i mod n)th link. Suppose that we route xn commodities. The probability that a particular bin does not overflow is roughly $1 - x^2p^2 = 1 - x^2/n$. The probability that no link overflows is roughly $(1 - x^2/n)^n$. This becomes small as x becomes larger, and in particular, is negligible for a super-constant x. So we expect to admit $O(n)$ commodities before a link overflows.

Note that the adaptivity gap remains large even if we allow a large $D_{\max}$-$c_{\min}$ gap. Suppose that each link has capacity $1/\alpha - 1$ instead of 1. Then we can modify the above argument appropriately to obtain a gap of $O(n^\alpha)$. In particular, we take $p = 1/n^\alpha$. Then, as before, the adaptive optimum gets value roughly $n(1/\alpha - 1)/p$. On the other hand, by our rough argument, when the non-adaptive solution admits xn commodities, the probability of overflow is roughly $(1 - x^{1/\alpha}p^{1/\alpha})^n = (1 - x^{1/\alpha}/n)^n$, which is negligible if $x = \omega(1)$. Therefore the non-adaptive solution admits $O(n)$ commodities in expectation, and we get a gap of $\Omega(n^\alpha)$.

Next, we prove that the natural randomized rounding algorithm for this problem obtains a matching approximation to the optimal adaptive policy.

In particular, consider the optimal solution f to $LP(I, \frac{1}{en^{2\alpha}}c)$, and let $\{f_P^{(i)}\}$ be any path-decomposition for it. We will bound the performance of the following non-adaptive strategy: for all i, pick a path $P \in \mathcal{P}_i$ with probability $f_P^{(i)}$ and route commodity i along this path.

We bound the probability that *any* edge in the graph contains more flow than its capacity. Then, the expected value of our solution is at least one minus that probability times the optimal value of $LP(\frac{1}{en^{2\alpha}}c, I)$.

Consider any edge e and let p_i be the probability that we route commodity i along this edge. Note that $\sum_i p_i\mu_i \le \frac{1}{en^{2\alpha}}c_e$. Let the random variable Y_i denote the flow of commodity i along this edge, and let $X_i = Y_i/(\alpha c_e)$. Note that $X_i \in [0,1]$. Furthermore, $\mu = \sum_i \mathbf{E}[X_i] = \sum_i \frac{p_i\mu_i}{\alpha c_e} \le \frac{1}{en^{2\alpha}\alpha}$. Using Chernoff bounds [14],

$$\mathbf{Pr}\left[\sum_i Y_i \ge c_e\right] = \mathbf{Pr}\left[\sum_i X_i \ge 1/\alpha\right] \le \left(\frac{e\mu}{1/\alpha}\right)^{1/\alpha} \le \left(n^{-2\alpha}\right)^{1/\alpha} = 1/n^2$$

Taking a union bound over all edges in the graph (at most $n^2/2$ of them), we get that the probability of failure is at most $1/2$. The expected value of the solution is therefore at least half the value of $LP(\frac{1}{en^{2\alpha}}c, I)$, which by Corollary 3.2 is at least a $\frac{1}{4en^{2\alpha}}$ fraction of the value of the optimal adaptive solution.

E Routing with Congestion

The approximation factor achieved by our algorithm scales inversely with $1 - \alpha$, where α is the ratio between the maximum possible demand and the minimum capacity in the network. A nice feature of our algorithm is that if it is allowed to exceed the capacity of any link by a small factor, its performance improves considerably.

In particular, if the algorithm is allowed to exceed capacities by a factor of $(1+\epsilon)$, we run the algorithm IR using $\hat{c}_e = (1-\alpha+\epsilon)c_e$ for each $e \in E$, instead of $(1-\alpha)c_e$. Then Corollary 3.2 implies that the dependence of our approximation guarantees on α improves from $\frac{1}{1-\alpha}$ to $\frac{1}{1-\alpha+\epsilon}$. In particular, with a factor of 2 congestion, we obtain approximation guarantees independent of α.

Note that allowing the algorithm excess capacity does not make the problem simpler from the point of view of non-adaptive algorithms. The example in Appendix D above provides an adaptivity gap of $\Omega(n^{1/(2+\epsilon)})$ even if we allow augmentation up to a $(1+\epsilon)$ factor.