Single-Source Stochastic Routing

Shuchi Chawla[1*] and Tim Roughgarden[2**]

[1] Microsoft Research, Silicon Valley Campus, Mountain View, CA 94043
shuchi@cs.wisc.edu
[2] Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305 tim@cs.stanford.edu

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.

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 $(s_1, t_1), \dots, (s_k, t_k)$ 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 [5, 6] 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

* 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.
** Supported in part by ONR grant N00014-04-1-0725, DARPA grant W911NF-05-1-0224, and an NSF CAREER Award.

governed by a distribution with known mean μ_i, and that the sizes of different demands are independent. We also assume that there is a 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 [4–6, 8]. We show in the full version of this paper [2] that the problems we consider have a large (polynomial) adaptivity gap, even in networks of parallel links. 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 successfully routed commodities.

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 may 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 design *safe* routing policies, but bound their performance relative to an optimal *unsafe* routing policy. This goal is somewhat analogous to previous work [5, 6] that designed non-adaptive policies with expected value close to that of 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 roughly $(1 - \alpha)/\epsilon$

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 capture only 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 single-sink unsplittable flow problem [7, 13, 14]. When this gap α is $O(1/\log n)$, even the general multicommodity stochastic routing problem can be approximated to within a constant factor using a straightforward randomized rounding algorithm. (See the full version [2] for details.) Our goal will be to design routing policies that have good (constant or logarithmic) approximation ratios for every fixed constant 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 [7, 13, 14]. 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 the *Single-Source Stochastic Routing (SSSR)* problem.

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), the optimal value of which is an upper bound on the expected value of an optimal (unsafe) routing policy, to guide the commodity and route selection at each stage. The algorithm 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 value of the LP, as our algorithm routes and instantiates demands.

As noted above, previous work on related stochastic optimization problems [4–6, 8] 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 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.) 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.

Related Work. Starting with the work of Dantzig [3] in 1955, stochastic optimization problems have been studied extensively in Operations Research (see e.g. [1, 18]). Owing to the complexity of optimally solving[3] general stochastic problems, much of this work has focused on the special cases of stochastic linear programming and *k-stage recourse* problems. Several recent works by the theoretical CS community have studied the recourse model. Starting with [12, 15], 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. [9–11, 17]). Some of this work has been extended to the k-stage versions of these problems [10, 16], albeit with approximation factors that depend linearly or even exponentially on k.

The work that is most closely related to ours is that of Dean, Goemans and Vondrak [5, 6, 8]. 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, and 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 [5, 6], 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 [5, 6], 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 [5, 6], 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 $c : E \rightarrow \Re^+$. We are given k commodities indexed by $i \in I$, each with a source-sink pair (s_i, t_i) and a value v_i. In Section 4, we will assume that all commodities share a common source s. The "size" or demand of a commodity i is given by the random variable

[3] The optimal solution to a stochastic optimization problem such as SSSR can be a complex, exponential-size decision diagram. The number of possible solutions can be doubly-exponential in the number of stages.

D_i, drawn from an independent distribution with mean $\mu_i = \mathbf{E}[D_i]$. 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

An LP Relaxation for the Optimal Routing Policy. We now 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. We begin with a linear program giving an upper bound on the expected value of an optimal (unsafe) routing policy for a given stochastic routing instance:

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

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 1. *The expected value obtained by an optimal adaptive 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 1 is similar to a result by Dean, Goemans, and Vondrak [6] in the special case of a single-link network (Knapsack). Scaling, we also obtain the following corollary.

Corollary 1. *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 $\alpha = D_{max}/c_{min}$.*

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 $\hat{f}$.

Fig. 1. High-level description of the algorithm IR.

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

Fact 1 *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 next discuss the much more subtle issue of path selection. 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 1. *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 $q \le r$ 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. For example, in a stochastic routing instance in a single-link network (i.e., Knapsack), every flow path is 1-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 1. *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 1), 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 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 flow of every edge $e \in P^*$ has decreased by at least $\min\{\hat{f}_e^{(-j)}, d_j\}$. 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 define a flow g feasible for $LP(I^j, c^j)$.

We now elaborate on the first step. Initialize $g_P^{(i)}$ to $\hat{f}_P^{(i)}$ 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 $\hat{f}_e^{(-j)} - g_e^{(-j)} < \min\{\hat{f}_e^{(-j)}, d_j\}$. While $F^* \neq \emptyset$, we decrease flow on paths of $\mathcal{S}$ as follows. Consider the paths P of $\mathcal{S}$ with $g_P^{(i)} > 0$, $i \neq j$, 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 $\hat{f}_e^{(-j)} - g_e^{(-j)} = \min\{\hat{f}_e^{(-j)}, d_j\}$ 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 $\sum_{i \in I_j} \sum_{P \in \mathcal{P}_i} (\hat{f}_P^{(i)} - g_P^{(i)}) \leq r \cdot d_j$. 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 1 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. By the Principle of Deferred Decisions, the random variables X_i and D_i are independent for each i. Taking expectations in (4), we have

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

$$= (r+1)\sum_{i=1}^{k} v_i\,\mathbf{E}[X_i], \tag{6}$$

where (5) follows from the independence of X_i and D_i. Since Algorithm GREEDY-IR is a safe routing policy (Fact 1), the sum on the right-hand side of (6) is precisely the expected value obtained by Algorithm GREEDY-IR. □

To usefully apply Lemma 1, 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 1, which will be useful in Section 4.

Lemma 2. *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 $W = w_1/w_k$.*

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 the full version [2] for 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).

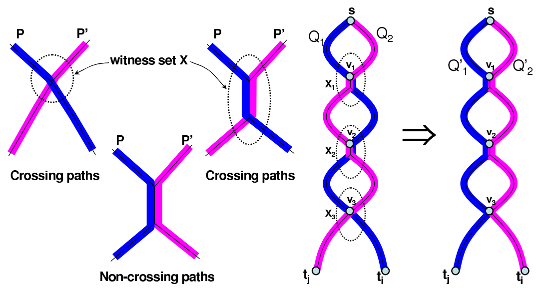
We obtain a constant-factor approximation in the special case where all of the sinks lie on a common face in some embedding of the planar network. Here, we show that every commodity has a 2-coverable path (Section 4.2). Lemma 1 then implies that the GREEDY-IR algorithm achieves a constant-factor approximation 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 $g : V \to \Re^2$ 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 2. *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 (Figure 2(a)). The set X is called a* witness *to this crossing of P and P'.*



(a) (b)

Fig. 2. (a) Crossing and non-crossing paths; (b) Converting a crossing path-decomposition to a non-crossing one

Definition 3. *A set of paths is said to be* non-crossing *if every pair of paths is distinct and non-crossing.*

Given two crossing paths, we can "uncross" them (Figure 2(b)). We therefore get the following lemma (proof omitted for brevity).

Lemma 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 intersections 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 in 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 (not including $g(t_i)$). We call this set the *region enclosed by i*.

Definition 4. *A commodity i* dominates *a commodity j if $g(t_i) \in \mathcal{A}_j$.*
It is easy to verify that the dominance relation defines a partial order on commodities.

Lemma 4. *If i dominates j, then $\mathcal{A}_i \subset \mathcal{A}_j$.*

Corollary 2. *The dominance relation is transitive and 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 5. *Let $\{f_P^{(i)}\}_{P \in \mathcal{S}}$ be a non-crossing path decomposition of the planar, single-source multicommodity flow f. Let i be an undominated commodity. Then every commodity i flow path in $\mathcal{S}$ is 2-covered by $\{f_P^{(i)}\}_{P \in \mathcal{S}}$.*

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

Lemmas 2 and 5 easily imply a constant-factor approximation ratio for the GREEDY-IR algorithm when all sinks like on a common face in some planar embedding. In particular, if we consider a planar embedding of the graph with all sinks on the outer face, then by definition, all the commodities are undominated.

Theorem 2. *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.*

Of course, Theorem 2 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, we show in the full version that the commodity with the highest value of w_i may not be $o(\log k)$-coverable. However, as we show below, having at least one 2-coverable commodity in every 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: $I_x = \{i : w_i \in [2^x, 2^{x+1})\}$ for each $x \in \{0, \cdots, \log W\}$.

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 any undominated commodity and route it along a flow path in a non-crossing path decomposition of the flow.

Theorem 3. *Algorithm* PLANAR-IR *is a* $(5\frac{1+\alpha}{1-\alpha}\log W)$*-approximation.*

Proof. Since we pick the best over $\log W$ groups of commodities, $\mathcal{V}_{x^*}$ is at least a $1/\log W$ fraction of the value of $LP(I,(1-\alpha)c)$. Now Lemma 5 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 2 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 obtained by an optimal routing policy for all of the demands. □

References

1. J. R. Birge and F. Louveaux. *Introduction to stochastic programming*. Springer Series in Operations Research. Springer-Verlag, New York, 1997.
2. S. Chawla and T. Roughgarden. Single-source stochastic routing. http://www.cs.cmu.edu/∼shuchi/papers/stoch-routing.ps.
3. G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.
4. B. Dean. *Approximation Algorithms for Stochastic Scheduling Problems*. PhD thesis, Massachusetts Institute of Technology, Massachusetts, 2005.
5. B. Dean, M. Goemans, and J. Vondrak. Adaptivity and approximation for stochastic packing problems. In *SODA '05*, pages 395–404.
6. B. Dean, M. Goemans, and J. Vondrak. The benefit of adaptivity: Approximating the stochastic knapsack problem. In *FOCS '04*, pages 208–217.
7. Y. Dinitz, N. Garg, and M. X. Goemans. On the single-source unsplittable flow problem. *Combinatorica*, 19(1):17–42, 1999.
8. M. Goemans and J. Vondrak. Stochastic covering and adaptivity. In *LATIN '06*, pages 532–543.
9. A. Gupta, M. Pal, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *STOC '04*, pages 417–426.
10. A. Gupta, M. Pal, R. Ravi, and A. Sinha. What about wednesday? approximation algorithms for multistage stochastic optimization. In *APPROX '05*.
11. A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In *FOCS '04*, pages 218–227.
12. N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *SODA '04*, pages 684–693.
13. J. Kleinberg. Single-source unsplittable flow. In *FOCS '96*, pages 68–77.
14. S. G. Kolliopoulos and C. Stein. Approximation algorithms for single-source unsplittable flow. *SIAM Journal on Computing*, 31(3):919–946, 2001.
15. R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. *Mathematical Programming*, 2005.
16. D. B. Shmoys and C. Swamy. Sampling-based approximation algorithms for multistage stochastic optimization. In *FOCS '05*.
17. D. B. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *FOCS '04*, pages 228–237.
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.