

Congestion minimization for multipath routing via multiroute flows

Chandra Chekuri^{*1} and Mark Idleman^{†2}

1 Dept. of Computer Science, University of Illinois, Urbana, 61801, USA
chekuri@illinois.edu

2 Dept. of Computer Science, University of Illinois, Urbana, 61801, USA
middleman2@illinois.edu

Abstract

Congestion minimization is a well-known routing problem for which there is an $O(\log n / \log \log n)$ -approximation via randomized rounding [17]. Srinivasan [18] formally introduced the low-congestion *multi-path* routing problem as a generalization of the (single-path) congestion minimization problem. The goal is to route multiple disjoint paths for each pair, for the sake of fault tolerance. Srinivasan developed a *dependent* randomized scheme for a special case of the multi-path problem when the input consists of a given set of disjoint paths for each pair and the goal is to select a given subset of them. Subsequently Doerr [7] gave a different dependent rounding scheme and derandomization. In [8] the authors considered the problem where the paths have to be chosen, and applied the dependent rounding technique and evaluated it experimentally. However, their algorithm does not maintain the required disjointness property without which the problem easily reduces to the standard congestion minimization problem.

In this note we show a simple algorithm that solves the problem correctly without the need for dependent rounding — standard independent rounding suffices. This is made possible via the notion of multiroute flows originally suggested by Kishimoto et al. [13]. One advantage of the simpler rounding is an improved bound on the congestion when the path lengths are short.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems: Computations on Discrete Structures

Keywords and phrases multipath routing, congestion minimization, multiroute flows

Digital Object Identifier 10.4230/OASICS.SOSA.2018.3

1 Introduction

Congestion minimization is a routing problem which originally arose in the context of wire routing problem in VLSI design. It is also a relaxation of the classical disjoint paths problem. Here we restrict attention to directed graphs. The input to these problem consists of a directed graph $G = (V, E)$ and a collection of source-sink pairs $(s_1, t_1), (s_2, t_2), \dots, (s_h, t_h)$. The edge-disjoint paths problem (EDP for short) is the following: given the graph and the pairs, can the given pairs be connected via edge-disjoint paths? More formally, are there edge-disjoint paths $P_1 \dots, P_h$ such that for $1 \leq i \leq h$, P_i is an s_i - t_i path? This is a classical NP-Complete problem. An optimization related to EDP is the *congestion minimization* problem. The goal is to find a collection of paths for the given pairs such that the *congestion* of the paths is minimized. The congestion of an edge e with respect to a collection of paths is

* Supported in part by NSF grants CCF-1319376 and CCF-1526799.

† Part of this work appeared in the author's recent MS thesis [11].



the number of paths in the collection that contain e , and the congestion of a given collection of paths is simply the maximum congestion over all the edges. Raghavan and Thompson, in their influential work [17], introduced the randomized rounding technique and obtained an $O(\log n / \log \log n)$ approximation via a natural multicommodity flow relaxation. Here n is the number of nodes in the graph. Surprisingly this approximation ratio was recently shown to be the right threshold of approximability [6] (modulo appropriate complexity theoretic assumption). There are generalizations of the congestion minimization problem where the pairs have demands and the edges have capacities. We restrict attention to the basic version with unit demands and unit capacities. The results for this basic version generalize easily.

Multipath routing for fault tolerance: The focus of this paper is *multipath* routing. This is motivated by fault tolerance considerations in high-capacity networks such as optical networks. In such networks each pair (s_i, t_i) needs to be connected via k_i disjoint paths; in typical applications $k_i = 2$. The idea is to protect the connection between s_i and t_i in case of edge or node failures. We will restrict our attention to edge failures since node failures can be addressed via appropriate reductions in directed graphs. In the network literature the case of $k_i = 2$ is typically referred to a 1 + 1 or 1 : 1 protection. See [9] for some relevant background and additional references.

Srinivasan [18] considered approximation algorithms for the multipath congestion minimization problem that he formalized as follows. The input is a directed graph $G = (V, E)$ and h source-sink pairs as before. In addition, for each (s_i, t_i) , we have an integer requirement $k_i \geq 1$. The goal is to choose for each pair i a set of k_i edge-disjoint s_i - t_i paths $Q_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,k_i}\}$ so as to minimize the congestion induced by the collection of paths $\mathcal{Q} = \cup_i Q_i$. Srinivasan developed an $O(\log n / \log \log n)$ -approximation for a variant of this problem. He assumes that the input includes for each i a collection of disjoint paths \mathcal{P}_i . The goal now is to choose for each i a sub-collection of exactly k_i paths from \mathcal{P}_i so as to minimize the congestion of the chosen paths. For this purpose Srinivasan developed and used his influential *dependent rounding* technique for cardinality constraints [18]. We refer the reader to [9, 7, 5] for several subsequent developments on dependent rounding including derandomization, and the ability to handle more general constraints than cardinality constraints. Doerr et al. [8] consider the multipath congestion minimization problem where the path collection \mathcal{P}_i is not explicitly given and the goal of the algorithm is to find k_i disjoint paths for each pair i so as to minimize the congestion of the chosen path collection. They write a natural multicommodity flow LP relaxation and use dependent rounding techniques to derive an $O(\log n / \log \log n)$ congestion bound. However, their algorithm does not maintain the crucial property that the chosen path collection for each pair i is edge disjoint! If there is no requirement of edge-disjointness, the problem is easily reduced to the standard congestion minimization problem: simply create k_i copies of each pair (s_i, t_i) with only one path for each copy required.

Our contribution: This note is motivated by two considerations. The first is to address the deficiency of the algorithm from [8] that we pointed out. We also note that the assumption that the input consists of a large number of disjoint paths for each pair, as assumed by Srinivasan, may not be the right model. Indeed, in practice a number of candidate paths are generated for each pair but they need not all be disjoint. Typically, the edge connectivity in high-speed networks is not very large. Can we solve the multipath congestion minimization problem when the paths are not explicitly given? Our second motivation is regarding the use of dependent rounding. Dependent rounding has been an elegant and powerful methodology with several new applications. Nevertheless, it is useful to examine whether specific applications really need it since the dependent rounding adds to the complexity of

the algorithm from a conceptual and implementation point of view.

Here we show that there is a simple solution to the multipath congestion minimization problem via the notion of multiroute flows. We show that standard randomized rounding can be used to obtain an $O(\log n / \log \log n)$ approximation. In other words there is no need to use dependent rounding for the multipath congestion minimization problem. There is also a concrete advantage to the simpler rounding. It allows us to improve the approximation to $O(\log d / \log \log d)$ when the paths have only d edges; this improvement requires the use of the Lovasz local lemma (LLL) and its constructive version [15, 10]. As far as we are aware this improvement is not easy to obtain via the dependent rounding approach.

One of our goals is to highlight multiroute flows. These flows were originally introduced by Kishimoto et al. [13] and some of their properties have been clarified by Aggarwal and Orlin [1]. Multiroute flows are a useful concept when considering fault-tolerance in networks, and it appears that they are less well known in the theoretical computer science literature although there have been several previous applications [3, 14, 4].

Finally, the simpler algorithm presented here inspired the following algorithmic question that we briefly describe although it is not the main focus of this paper. Given an s - t flow in a directed graph $G = (V, E)$ it is well-known that it can be decomposed into paths and cycles in $O(nm)$ time where $n = |V|$ and $m = |E|$. The corresponding question for multiroute flows has not been explored systematically. Given an s - t k -route flow (see Section 2 for definitions and background on multiroute flows) how fast can we decompose it into a collection of elementary k -flows? The existence of a polynomial time algorithm follows from the properties of these flows [1]. The MS thesis of the second author [11] describes faster exact and approximation algorithms for the multiroute flow decomposition problem.

2 Background on multiroute flows

We will assume that graphs are directed for the rest of the paper. Unless otherwise noted, we will use n and m to represent the number of nodes and edges of a graph in question, respectively.

Network flow and flow decomposition: Given a directed graph $G = (V, E)$ with non-negative capacities c_e on each edge $e \in E$, a *network flow* is defined as a function $f : E \rightarrow \mathbb{R}^+$. We will often express flows as vectors to help differentiate them from scalar values; for a given flow \vec{f} written in this way, f_e is the flow value on an edge e . A *feasible flow* of k units from a source node s to a target node t is a flow \vec{f} such that

1. The flow value f_e on each edge $e \in E$ satisfies $0 \leq f_e \leq c_e$,
2. k units of flow leave vertex s and enter vertex t , i.e.,

$$\sum_{e=(s,u)} f_e - \sum_{e=(u,s)} f_e = \sum_{e=(u,t)} f_e - \sum_{e=(t,u)} f_e = k, \text{ and}$$

3. \vec{f} satisfies the *flow conservation property*: for each vertex $v \in V \setminus \{s, t\}$,

$$\sum_{e=(u,v)} f_e = \sum_{e=(v,u)} f_e.$$

The above definition characterizes network flows based on the flow value assigned to each edge in the graph; we will refer to such formulations as *edge-based flows*. It is often beneficial to also consider an equivalent formulation of network flows, the *path-based flow*. Let \mathcal{P}_{st} be the set of all paths from s to t . In the path based formulation, an s - t flow in

a graph G is defined as a function $f : \mathcal{P}_{st} \rightarrow \mathbb{R}_+$. Although this definition works with an implicit and exponential sized set \mathcal{P}_{st} , there are several advantages to this view. The path-based and edge-based formulations of s - t flows are “equivalent” in a certain sense. One can see that a path-based flow induces an edge-based flow of the same value, and conversely, *flow-decomposition* allows one convert an edge-based acyclic flow into a path-based flow of the same value (note that the decomposition is not unique). Such a decomposition can be computed in $O(nm)$ time.

Multiroute aka k -route flows: Let k be a non-negative integer. Given a directed graph $G = (V, E)$ and two distinct nodes $s, t \in V$, an *elementary k -flow* from s to t is defined as an s - t flow of k total units, consisting of 1 unit sent along each of k edge-disjoint s - t paths. Equivalently we can view an elementary k -flow as a tuple (p_1, p_2, \dots, p_k) where each $p_i \in \mathcal{P}_{st}$ and the paths p_1, p_2, \dots, p_k are mutually edge-disjoint. Let $\mathcal{P}_{st}^{(k)}$ be the set of all elementary k -flows from s to t in G . We will some times use the notation \bar{p} to denote an elementary k -flow in $\mathcal{P}_{st}^{(k)}$. A k -route s - t flow is defined as $f : \mathcal{P}_{st}^{(k)} \rightarrow \mathbb{R}_+$, in other words, as a non-negative sum of elementary k -flows. The value of a k -route flow f is simply $\sum_{\bar{p} \in \mathcal{P}_{st}^{(k)}} f(\bar{p})$. A k -route flow f is feasible if the total flow on any edge is at most its capacity $c(e)$; that is, $\sum_{\bar{p} \ni e} f(\bar{p}) \leq c(e)$ for each e . Note that the case of $k = 1$ is simply the standard definition of flow via the path formulation.

Two natural questions arise. Can a maximum k -route flow be computed efficiently? Second, what is the relationship between standard flows and k -route flows. The first question is easy to answer. One can write a natural LP relaxation for the maximum k -route flow based on the definition, however, the number of variables is $|\mathcal{P}_{st}^{(k)}|$ and hence exponential in the graph size. However, the number of non-trivial constraints is only m and one see that the separation oracle for the dual LP is poly-time solvable via min-cost flow. However, there are much faster algorithms via a crucial property that connects k -route flows to standard flows (1-route flows). We have defined k -route flows via a path formulation. We say that an edge-based flow $f : E \rightarrow \mathbb{R}_+$ is a k -route flow if it can be decomposed into a k -route flow. More formally f is a k -route flow if there is a $g : \mathcal{P}_{st}^{(k)} \rightarrow \mathbb{R}_+$ such that $f(e) = \sum_{\bar{p} \ni e} g(\bar{p})$.

The following theorem, first proved by Kishimoto [12] and subsequently simplified by Aggarwal and Orlin [1], gives a simple necessary and sufficient condition for a flow to be a k -route flow. See Fig 1. This condition is related to the integer decomposition property of polytopes defined by totally unimodular matrices [2].

► **Theorem 1** ([12, 1]). *An acyclic edge-based s - t flow $f : E \rightarrow \mathbb{R}_+$ can be decomposed into an s - t k -route flow if and only if $f(e) \leq v/k$ for each $e \in E$, where v is the value of f .*

The proof of the preceding theorem gives a polynomial time algorithm for the decomposition. Recently we have improved the running time for the decomposition; details can be found in [11].

Aggarwal and Orlin describe an algorithm that finds a maximum s - t k -route flow via $\min\{k, \log(kU)\}$ standard maximum flow computations. Here U is the maximum capacity of any edge in the graph. Note that the algorithm returns an edge-based multiroute flow.

3 Multipath routing via multiroute flows

In this section we consider the multipath minimum congestion routing problem via multiroute flows. We are given a directed network $G = (V, E)$ along with h commodities, each of which consists of a pair of vertices (s_i, t_i) in G . For each commodity (s_i, t_i) , the node s_i is referred to as the *source* node, and the node t_i is referred to as the *sink* node. For each commodity

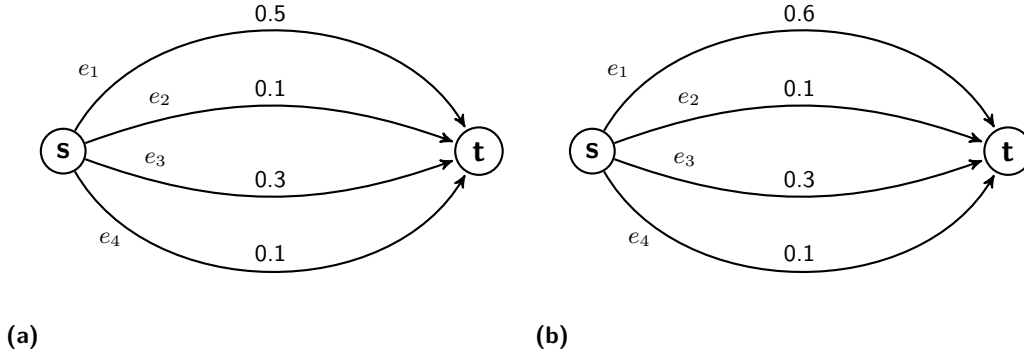


Figure 1 The flow in (a) is decomposable into a 2-route flow by sending 0.3 units of flow along e_1 and e_3 , 0.1 units of flow along e_1 and e_2 , and 0.1 units of flow along e_1 and e_4 . The flow in (b) is not decomposable into a 2-route flow; the 0.6 units of flow on edge e_1 presents a bottleneck to obtaining a valid decomposition.

we are also given an integer k_i which is the number of edge-disjoint paths needed for pair i . For notational simplicity we will assume that $k_i = k$ for all i . It is easy to generalize the entire approach when k_i are different. The goal is to find for each i an elementary k -flow $\bar{p}_i \in \mathcal{P}_{st}^{(k)}$ such that the congestion induced by the path collection in $\cup_i \bar{p}_i$ is minimized. The whole approach can also be generalized to the setting where pairs have demands and edges have capacities and the goal is to minimize the relative congestion. We avoid this general version for notational simplicity.

As we mentioned, Srinivasan [18, 9] considered the version of the problem where the elementary k -flow for each i has to be chosen from a given set of disjoint path collection $\mathcal{P}_i \subseteq \mathcal{P}_{st}$. Here we consider the version where the algorithm is not given such a path collection. (We later show that the given paths case can be treated as a special case.) We write two different relaxations, one based on the path based definition of multiroute flows and the other based on the edge based definition (via Theorem 1).

Fig 2 describes the path-based formulation. For ease of notation we will assume that the pairs are distinct (otherwise we can add dummy terminals to achieve this) and hence $\mathcal{P}_{s_i t_i}^{(k)}$ for the different i are distinct. For each $\bar{p} \in \cup_i \mathcal{P}_{s_i t_i}^{(k)}$ we have a flow variable $x(\bar{p})$ indicating the amount of flow that is sent on the elementary k -flow \bar{p} . The natural constraints are that the total k -route flow for (s_i, t_i) is 1 for each i . The goal is to minimize the maximum flow on any edge e which is the variable C . This is the same as minimizing the maximum congestion since we are working with the case when all capacities are 1.

Fig 3 describes the edge-based formulation. Here we have variables $x_{i,e}$ to indicate the flow for pair i on edge e . In addition to flow conservation constraints we seek a total flow of k units from s_i to t_i . The goal is again to minimize the maximum flow on any edge e which is the variable C . Note that we include the constraint that $x_{i,e} \leq 1$ for each e and each i . This is crucial. If we did not include this constraint we would not be able to guarantee that the flow for pair i defined by the variables $x_{i,e}$ is a k -route flow. We observe that [8] write this same relaxation.

The proof of the following lemma easily follows from Theorem 1. We note that the constraint $x_{i,e} \leq 1$ is necessary.

► Lemma 2. *The two LP relaxation are equivalent in that any feasible solution to one can be used to define a corresponding feasible solution of the same or better value to the other.*

3:6 Multipath Routing via Multiroute Flows

$$\begin{aligned}
& \min C \\
& \sum_{\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}} x(\bar{p}) = 1 \quad i = 1, \dots, h \\
& \sum_{i=1}^h \sum_{\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}, e \in \bar{p}} x(\bar{p}) \leq C \quad e \in E \\
& x(\bar{p}) \geq 0 \quad \bar{p} \in \cup_i \mathcal{P}_{s_i t_i}^{(k)}
\end{aligned}$$

■ **Figure 2** LP relaxation for multipath congestion minimization via path-based flows.

$$\begin{aligned}
& \min C \\
& \sum_{e=(w,v) \in E} x_{i,e} - \sum_{e=(v,w) \in E} x_{i,e} = 0 \quad v \in V \setminus \{s_i, t_i\}, i = 1, \dots, h \\
& \sum_{e=(s_i,v) \in E} x_{i,e} - \sum_{e=(v,s_i) \in E} x_{i,e} = k \quad i = 1, \dots, h \\
& \sum_{i=1}^h x_{i,e} \leq C \quad e \in E \\
& x_{i,e} \in [0, 1] \quad e \in E, i = 1, \dots, h
\end{aligned}$$

■ **Figure 3** LP relaxation for multipath congestion minimization via edge-based flows.

Proof. We sketch the more interesting direction. Consider a feasible solution the edge-based LP. Consider any commodity i and the s_i - t_i flow of k units induced by the variables $x_{i,e}$. Since $x_{i,e} \leq 1$ for each e this flow satisfies the conditions of Theorem 1 and hence can be decomposed into a path-based k -route flow of value 1. We do this decomposition for each i to generate a path-based flow solution. It is easy to see that it is feasible for the path-based LP. ◀

We observe that the path-based LP can be solved in polynomial time. There are two ways to see this. One way is to note that the separation oracle for the dual of the path-based LP is min-cost flow. The other way is via the equivalence shown in Lemma 2 since the edge-based LP is a polynomial sized formulation; one can decompose the edge-based flow for each pair i into a path based flow via Theorem 1.

In [8] the following rounding strategy is used. They first solve the edge-based flow LP. Let C^* be the congestion in the fractional solution. We will assume without loss of generality that $C^* \geq 1$ since we know that 1 is a lower bound on the optimum integral solution. For each commodity i the $x_{i,e}$ variables define a flow of value k . They do a standard flow decomposition of this flow to obtain a collection of paths $\mathcal{Q}_i = \{p_{i,1}, \dots, p_{i,\ell_i}\} \subset \mathcal{P}_{s_i t_i}$ and associated fractions $\alpha_{i,1}, \dots, \alpha_{i,\ell_i}$ such that $\sum_j \alpha_{i,j} = k$ and $0 \leq \alpha_{i,j} \leq 1$. Then they apply the dependent rounding scheme of Srinivasan or the variant developed in [7] to select exactly k paths from \mathcal{Q}_i . They do this process independently for each commodity i . They exploit the negative correlation properties of the dependent rounding which implies that the process behaves as if the paths are chosen independently and hence one can apply Chernoff-bound

style analysis and the union bound. This allows one to show that the congestion obtained by the rounding is, with high probability, $O(\log n / \log \log n \cdot C^*)$. One can, via well-known Chernoff-inequalities, also show related bounds that provide improved bounds when C^* is large.

The main issue with the above rounding is the fact that the path collection \mathcal{Q}_i obtained via standard flow-decomposition is not guaranteed to give a collection of disjoint paths. Thus, the k paths chosen for (s_i, t_i) may not in fact be edge disjoint.

Randomized rounding via k -route flows: It is convenient to describe our rounding algorithm via the path-based LP formulation. We solve the LP to find a fractional solution with congestion C^* . Note that the LP solution gives us for each i a k -route flow of value 1. More formally for each i we have a collection $\mathcal{Q}_i = \{\bar{p}_{i,1}, \dots, \bar{p}_{i,\ell_i}\}$ where $\bar{p}_{i,j} \in \mathcal{P}_{s_i t_i}^{(k)}$ for each j , and also associated non-negative numbers $\alpha_{i,1}, \dots, \alpha_{i,\ell_i}$ such that $\sum_j \alpha_{i,j} = 1$. Note that we have a convex combination over elementary k -flows for each i . Now we can perform a simple randomized rounding similar to what Raghavan and Thompson did for the standard congestion minimization problem. For each i , we independently pick a single elementary k -flow $\bar{p}_{i,j}$ from \mathcal{Q}_i where the probability of picking it is exactly $\alpha_{i,j}$. Since we are picking an elementary k -flow for each i , we are guaranteed that the k paths for each i are edge disjoint. We can use the same standard argument as in [17] to argue that the congestion induced by this randomized rounding is $O(\log n / \log n \cdot C^*)$. Note that we do not need to use any dependent rounding techniques since all the work has been done for us via the multiroute flow based LP relaxation.

For the sake of completeness we prove the desired bound on the congestion. We first state the standard Chernoff bound that we need (see [16]).

► **Theorem 3.** *Let X_1, \dots, X_n be n independent random variables (not necessarily distributed identically), with each variable X_i taking a value of 0 or v_i for some value $0 < v_i \leq 1$. Then for $X = \sum_{i=1}^n X_i$, $E[X] \leq \mu$, and $\delta > 0$,*

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

► **Lemma 4.** *With probability $1 - 1/\text{poly}(n)$, the congestion resulting from the randomized rounding algorithm is $O(\log n / \log \log n \cdot C^*)$.*

Proof. For each edge $e \in E$, define $X_{i,e}$ to be a binary random variable where $X_{i,e} = 1$ if e lies on one of the s_i - t_i paths making up the elementary k -flow chosen by the above randomized rounding scheme, and 0 otherwise. Note that $E[X_{i,e}] = x_{i,e}$ is the total flow on e for commodity i . Let $Y_e = \sum_{i \in [h]} X_{i,e}$ be the random variable which is the total number of paths using edge e in the chosen solution. Note that

$$E[Y_e] = \sum_{i \in [h]} E[X_{i,e}] = \sum_{i \in [h]} \sum_{j \in [\ell_i]: e \in \bar{p}_{i,j}} x(\bar{p}_{i,j}) = \sum_{i \in [h]} x_{i,e} \leq C^*.$$

In the above we use the fact that any edge e belongs only to one of the paths of an elementary k -flow since the paths making up the elementary k -flow are by definition edge disjoint. For any edge $e \in E$, the variables $X_{i,e}$, $i \in [h]$ are independent via the rounding procedure; therefore, the bound in Theorem 3 applies. Choose δ such that $(1 + \delta) = \frac{c \ln n}{\ln \ln n}$ for some constant c that will be determined later. Assume $n > e$ so that $\ln \ln n - \ln \ln \ln n > 0.5 \ln \ln n$.

3:8 Multipath Routing via Multiroute Flows

By letting $\mu = C^* \geq 1$ in Theorem 3, we then have

$$\begin{aligned}
 \Pr[Y_e \geq (1 + \delta)C^*] &< \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \right)^{C^*} \\
 &\leq \frac{e^\delta}{(1 + \delta)^{(1+\delta)}} \tag{*} \\
 &\leq \frac{e^{(1+\delta)}}{(1 + \delta)^{(1+\delta)}} \\
 &= \left(\frac{c \ln n}{e \ln \ln n} \right)^{(-c \ln n / \ln \ln n)} \\
 &= \exp((\ln c/e + \ln \ln n - \ln \ln \ln n)(-c \ln n / \ln \ln n)) \\
 &\leq \exp(0.5 \ln \ln n (-c \ln n / \ln \ln n)) \\
 &\leq \frac{1}{n^{c/2}}
 \end{aligned}$$

There are at most n^2 edges, so by the union bound, we have

$$\begin{aligned}
 \Pr \left[\max_{e \in E} Y_e \geq (1 + \delta)C^* \right] &\leq \sum_{e \in E} \Pr[Y_e \geq (1 + \delta)C^*] \\
 &\leq n^2 \cdot \frac{1}{n^{c/2}} = n^{2-c/2}.
 \end{aligned}$$

Choosing $c = 8$ makes the claim fail to hold with probability at most $\frac{1}{n^2}$, and ensures that the inequality marked with (*) above is true (this choice of c ensures that the value of the expression within parenthesis is less than 1). This probability can be made arbitrarily small by increasing c . Because

$$(1 + \delta)C^* = (c \ln n / \ln \ln n)C^* = O(\log n / \log \log n) \cdot C^*,$$

this completes the proof. ◀

Using variants of the Chernoff bounds we can prove the following two lemmas as well. These bounds show improved relative bounds on the congestion when C^* is large. One can also show similar analysis if the capacities are large compared to the demands. The analysis is standard and we omit details in this version.

► **Lemma 5.** *If $C^* \geq 1$, then for any δ with $0 \leq \delta \leq 1$, there exists some constant $c > 0$ such that the congestion resulting from the randomized rounding algorithm is no more than $(1 + \delta)C^* + c \log n / \delta^2$ with probability $1 - 1/n^{\Omega(c)}$.*

► **Lemma 6.** *There is a constant $c > 1$ such that if $C^* \geq c \ln n$, then with high probability, the congestion of the rounding algorithm is at most $C^* + \sqrt{C^*(c \ln n)}$.*

3.1 Short paths and improved congestion via local lemma

In this section we point to another advantage of the simple rounding that we described in the preceding section. Consider the basic congestion minimization problem when $k = 1$. It is known that if all the flow paths in the fractional solution are “short”, then the congestion bound improves. More formally, if all the paths in the decomposition have at most d edges then one can obtain an integral solution with congestion $O(\log d / \log \log d \cdot C^*)$ [19]. This can be substantially better than the bound of $O(\log n / \log \log n \cdot C^*)$ when $d \ll n$. One can

also ensure that the flow paths are short by solving a path-based LP relaxation with the restriction that the flow is only on paths with at most d edges. The rounding that achieves the improved bound relies on the Lovász local lemma. LLL based analysis does not yield a polynomial-time. Srinivasan [19] obtained a polynomial-time algorithm by derandomizing the LLL based algorithm. More recently, building on the constructive version of the LLL due to Moser and Tardos [15], Haupteler, Saha and Srinivasan [10] obtained a much simpler randomized polynomial time algorithm that achieves a congestion of $O(\log d / \log \log d \cdot C^*)$. In fact they consider a general class of min-max integer programs, and one can cast the single path routing problem as a special case after the flow-decomposition. In the general setting of min-max integer programs the structure of the routing problem is not relevant for the bound we seek. The only parameter that matters is the maximum number of constraints that any variable participates in, which corresponds to the length of the flow paths.

Now suppose we consider the multipath routing problem. The multiroute flow based path LP formulation can be easily seen to be a special case of min-max integer programs considered in [10]. We can thus obtain an improved congestion bound of $O(\log d / \log \log d \cdot C^*)$ where d is the maximum number of edges in any elementary k -flow. The following natural question then arises. Find a fractional multipath routing for the given instance with the additional constraint that for each pair (s_i, t_i) the elementary flow chosen has at most d edges. For single path setting this LP can be solved in polynomial time since the separation oracle for the dual LP can be seen to be the constrained shortest path problem which is polynomial-time solvable. Unfortunately the corresponding separation oracle for the multipath case is NP-Hard even when $k = 2$. Nevertheless, we can apply a simple trick to obtain a bi-criteria approximation. We can ensure that each elementary k -flow has at most $2d$ edges as follows.

We consider the following relaxation which adds additional constraints to the edge-based relaxation from Fig 3. The additional constraint says that for each commodity i , the total number of edges used is at most d in a fractional sense.

$$\begin{aligned}
 & \min C \\
 & \sum_{e=(w,v) \in E} x_{i,e} - \sum_{e=(v,w) \in E} x_{i,e} = 0 \quad v \in V \setminus \{s_i, t_i\}, i = 1, \dots, h \\
 & \sum_{e=(s_i,v) \in E} x_{i,e} - \sum_{e=(v,s_i) \in E} x_{i,e} = k \quad i = 1, \dots, h \\
 & \sum_{i=1}^h x_{i,e} \leq C \quad e \in E \\
 & \sum_{e \in E} x_{i,e} \leq d \quad i = 1, \dots, h \\
 & x_{i,e} \in [0, 1] \quad e \in E, i = 1, \dots, h
 \end{aligned}$$

■ **Figure 4** LP relaxation for multipath congestion minimization with additional constraint to limit the number of edges used in the flow to at most d .

We now describe the modification to the rounding algorithm. As before we consider each commodity i and consider the k -route flow given by the variables $x_{i,e}$. We decompose this into a path based flow to obtain a collection $\mathcal{Q}_i = \{\bar{p}_{i,1}, \dots, \bar{p}_{i,\ell_i}\}$ of elementary k -flows between s_i and t_i . Abusing notation, we let amount of flow on $\bar{p}_{i,j}$ as $x(\bar{p}_{i,j})$. Note that $\sum_j x(\bar{p}_{i,j}) = 1$. Let d_j be the number of edges in $\bar{p}_{i,j}$. From the constraint in the LP on

$\sum_e x_{i,e}$ we have that $\sum_j d_j x(\bar{p}_{i,j}) \leq d$. Let $\mathcal{Q}'_i \subseteq \mathcal{Q}_i$ be the subset of elementary k -flows such that each of them contains at most $2d$ edges. It is easy to see, via Markov's inequality, that $\sum_{j \in \mathcal{Q}'_i} x(\bar{p}_{i,j}) \geq 1/2$. By scaling up the fractional values of the elementary k -flows in \mathcal{Q}'_i by 2 we obtain a feasible solution to the path-based LP using only elementary k -flows which have at most $2d$ edges. We have thus found a feasible fractional solution to the path LP formulation with the following guarantees: (i) the congestion of the solution is at most $2C^*$ where C^* is the congestion of the original relaxation (ii) the support of the solution consists of elementary k -flows for each commodity i which have at most $2d$ edges. We can now apply the algorithm from [10] to round this solution to obtain a randomized algorithm with congestion $O(\log d / \log \log \cdot C^*)$. Note that there is nothing special about the factor of 2. We can obtain a trade off. For any $\epsilon > 0$ we can ensure that the elementary k -flows have at most $(1 + \epsilon)d$ edges while guaranteeing that the congestion is $O(\frac{1}{\epsilon} \cdot \log d / \log \log d \cdot C^*)$.

3.2 Choosing paths from a given collection

Now we consider the setting that Srinivasan considered in his original paper where he assumes that the input includes for each i , a collection of disjoint paths \mathcal{P}_i . The goal is to select exactly k paths from \mathcal{P}_i for each i . We can handle this problem also via multiroute flow decomposition. First, we consider the natural LP relaxation for this problem from [18]. For simplicity we will again assume that the given h pairs are distinct and hence $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for $i \neq j$. We have a variable $y(p)$ for each $p \in \cup_i \mathcal{P}_i$ to indicate whether p is chosen or not. We require k paths to be chosen from each \mathcal{P}_i and also that $y(p) \in [0, 1]$. Subject to these conditions we minimize the congestion. The relaxation is formally specified in Fig 5.

$$\begin{aligned}
 & \min C \\
 & \sum_{p \in \mathcal{P}_i} y(p) = k \quad i = 1, \dots, h \\
 & \sum_{i=1}^h \sum_{p \in \mathcal{P}_i, e \in p} y(p) \leq C \quad e \in E \\
 & y(p) \in [0, 1] \quad p \in \cup_i \mathcal{P}_i
 \end{aligned}$$

■ **Figure 5** LP relaxation for multipath congestion minimization when paths for each pair are specified.

Suppose we are given a feasible solution y to the preceding LP with congestion value C . We claim that we can find a feasible solution x to the LP in Fig 2 with congestion value at most C with the following additional condition: for any i , if $x(\bar{p}) > 0$ for some $\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}$ then \bar{p} is a tuple of k paths from \mathcal{P}_i . If this is true then we can do randomized rounding via x as before and obtain the desired congestion bound while picking for each i exactly k paths from \mathcal{P}_i . Moreover, if the paths in $\cup_i \mathcal{P}_i$ are short we can obtain an improved congestion bound via the algorithm described in the preceding subsection.

We now justify the claim. Suppose y is a feasible solution to the LP in Fig 5. Fix a particular i . Without loss of generality $\mathcal{P}_i = \{p_1, p_2, \dots, p_{\ell_i}\}$ for some $\ell_i \geq k$. Consider a graph H_i with two nodes s_i, t_i and ℓ_i parallel edges $e_1, e_2, \dots, e_{\ell_i}$ from s_i and t_i with unit capacities where e_j corresponds to the path p_j . We now create a flow of value k from s_i to t_i in H_i where the flow on edge e_j is equal to $y(p_j)$. Note that $y(p_j) \leq 1$ for each p_j . Thus, via

Theorem 1, we can decompose this flow into a s_i - t_i k -route flow in H_i . Suppose this k -route flow is given by $x' : \mathcal{Q}_{s_i t_i}^{(k)} \rightarrow [0, 1]$. Here $\mathcal{Q}_{s_i t_i}^{(k)}$ is the set of elementary k -flows in H_i ; each such elementary k -flow is a set of k distinct edges from $\{e_1, e_2, \dots, e_{\ell_i}\}$. For each $\bar{q} \in \mathcal{Q}_{s_i t_i}^{(k)}$ there is a unique $\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}$ where the edge e_j maps to the path p_j ; we set $x(\bar{p}) = x(\bar{q})$. We set $x(\bar{p}) = 0$ for all $\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}$ which don't correspond to an elementary k -flow in H_i . We do this for each i and the resulting x is the claimed feasible solution to the LP in Fig 2. By construction x satisfies (i) $\sum_{\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}} x(\bar{p}) = 1$ for each i and (ii) if $\bar{p} \in \mathcal{P}_{s_i t_i}^{(k)}$ and $x(\bar{p}) > 0$ then \bar{p} is a set of k paths from \mathcal{P}_i . It is also easy to check that the congestion induced by x on any edge is the same as the congestion induced by y .

References

- 1 Charu C. Aggarwal and James B. Orlin. On multiroute maximum flows in networks. *Networks*, 39(1):43–52, 2002. URL: <http://dx.doi.org/10.1002/net.10008>, doi:10.1002/net.10008.
- 2 Stephen Baum and Leslie E Trotter Jr. Integer rounding and polyhedral decomposition for totally unimodular systems. *Optimization and Operations Research*, 157:15–23, 1978.
- 3 Graham Brightwell, Gianpaolo Oriolo, and F Bruce Shepherd. Reserving resilient capacity in a network. *SIAM journal on discrete mathematics*, 14(4):524–539, 2001.
- 4 Chandra Chekuri, Alina Ene, and Ali Vakilian. Prize-collecting survivable network design in node-weighted graphs. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 98–109, 2012.
- 5 Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 575–584. IEEE, 2010.
- 6 Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, and Kunal Talwar. Hardness of routing with congestion in directed graphs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 165–178. ACM, 2007.
- 7 Benjamin Doerr. Randomly rounding rationals with cardinality constraints and derandomizations. *STACS 2007*, pages 441–452, 2007.
- 8 Benjamin Doerr, Marvin Künnemann, and Magnus Wahlström. Randomized rounding for routing and covering problems: Experiments and improvements. In *Proceedings of the 9th International Conference on Experimental Algorithms, SEA'10*, pages 190–201, Berlin, Heidelberg, 2010. Springer-Verlag. URL: http://dx.doi.org/10.1007/978-3-642-13193-61_7, doi:10.1007/978-3-642-13193-61_7.
- 9 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3):324–360, 2006.
- 10 Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the lovász local lemma. *Journal of the ACM (JACM)*, 58(6):28, 2011.
- 11 Mark Idleman. Approximation algorithms for the minimum congestion routing problem via k -route flows. Master's thesis, University of Illinois, July 2017.
- 12 Wataru Kishimoto. A method for obtaining the maximum multiroute flows in a network. *Networks*, 27(4):279–291, 1996. URL: [http://dx.doi.org/10.1002/\(SICI\)1097-0037\(199607\)27:4<279::AID-NET3>3.0.CO;2-D](http://dx.doi.org/10.1002/(SICI)1097-0037(199607)27:4<279::AID-NET3>3.0.CO;2-D), doi:10.1002/(SICI)1097-0037(199607)27:4<279::AID-NET3>3.0.CO;2-D.
- 13 Wataru Kishimoto and Masashi Takeuchi. m -route flows in a network. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 77(5):1–18, 1994. URL: <http://dx.doi.org/10.1002/ecjc.4430770501>, doi:10.1002/ecjc.4430770501.

- 14 Andrew McGregor and Bruce Shepherd. Island hopping and path colouring with applications to wdm network design. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 864–873. Society for Industrial and Applied Mathematics, 2007.
- 15 Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *Journal of the ACM (JACM)*, 57(2):11, 2010.
- 16 Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.
- 17 Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, December 1987. URL: <http://dx.doi.org/10.1007/BF02579324>, doi:10.1007/BF02579324.
- 18 A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, FOCS '01*, pages 588–, Washington, DC, USA, 2001. IEEE Computer Society. URL: <http://dl.acm.org/citation.cfm?id=874063.875563>.
- 19 Aravind Srinivasan. An extension of the lovász local lemma, and its applications to integer programming. *SIAM Journal on Computing*, 36(3):609–634, 2006.