Electronic Colloquium on Computational Complexity, Report No. 109 (2006)

Hardness of Directed Routing with Congestion

Julia Chuzhoy* Sanjeev Khanna†

August 28, 2006

Abstract

Given a graph G and a collection of source-sink pairs in G, what is the least integer c such that each source can be connected by a path to its sink, with at most c paths going through an edge? This is known as the *congestion minimization* problem, and the quantity c is called the *congestion*. Congestion minimization is one of the most well-studied NP-hard optimization problems. It is well-known that the elegant *randomized rounding* technique of Raghavan and Thompson can be used to obtain a solution with congestion at most $c^* + O(\frac{\log n}{\log\log n})$ where c^* is the optimal congestion. In this paper we show that there exists a $\delta > 0$ such that no polynomial-time algorithm can guarantee a solution with congestion $c^* + (\frac{\delta \log n}{\log\log n})$ unless NP is contained in ZPTIME$(n^{\log\log n})$.

We also study the directed edge-disjoint paths (EDP) problem with congestion. The input to this problem is a graph G and a collection of source-sink pairs in G, along with a congestion parameter c. The goal now is to route as many pairs as possible such that the congestion on each edge is strictly bounded by c. We show that for c ranging from 2 to $\Theta(\frac{\log n}{\log\log n})$, directed EDP with congestion is $n^{\Omega(1/c)}$-hard to approximate, even if the algorithm performance is compared to the optimal EDP solution with no congestion. We also give a very simple integrality gap construction that shows that the multicommodity-flow relaxation for directed EDP with congestion c has an integrality gap of $n^{\Omega(1/c)}$ for c ranging from 2 to $\Theta(\frac{\log n}{\log\log n})$. We note that it is known that this problem can be approximated to within $n^{O(1/c)}$ using the multicommodity-flow relaxation.

A surprising aspect of our hardness and integrality gap results for directed EDP with congestion is that the instances created have a *unique paths* property, namely, for each source-sink pair, there is a unique path connecting it in the instance. An immediate consequence of this property is that our hardness results also hold for the All-or-Nothing Flow problem where the requirement that each pair be routed along a single path is relaxed to sending a unit of flow between each pair. This is in a sharp contrast to the undirected setting where the All-or-Nothing flow problem is known to be approximable to within a poly-logarithmic factor. We note that our hardness results can be extended to the vertex versions of the respective problems, where congestion is measured on vertices and not edges of the input graphs. All our results hold on directed acyclic graphs.

*Laboratory for Computer Science, MIT, Cambridge, MA and Dept. of CIS, University of Pennsylvania, Philadelphia, PA. Email: `cjulia@csail.mit.edu`

†Dept. of CIS, University of Pennsylvania, Philadelphia PA. Email: `sanjeev@cis.upenn.edu`. Supported in part by an NSF Career Award CCR-0093117.

ISSN 1433-8092

1 Introduction

A fundamental optimization problem in network routing is the *congestion minimization* problem. The problem may be stated as follows: given an n-vertex graph $G(V, E)$ and a collection of source-sink pairs $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, find the minimum integer c, such that there exists a routing that connects each source s_i to its sink t_i by a path, with at most c paths using any edge. The quantity c is called the *congestion* of the solution. In general, the edges in graph G may have capacities associated with them, and the congestion of a solution is then defined to be the largest ratio between the number of paths using an edge and the capacity of the edge.

If we relax the routing for each source-sink pair to be a flow (instead of a path), we get a multi-commodity flow relaxation of the congestion minimization problem. This relaxation is well-known to be poly-time solvable via Linear Programming. A beautiful and celebrated result of Raghavan and Thompson [17] shows that if there exists a multicommodity flow routing with congestion c^*, then there exists an integral routing with congestion at most $c^* + O(\frac{\log n}{\log \log n})$. The technique used for converting a fractional routing into an integral one is remarkably simple. For each source-sink pair, decompose the flow into a collection of flow-paths. Now choose one of the flow paths to connect the source to the sink (integrally) with probability equal to the flow on the path. This technique, referred to as the *randomized rounding* technique, has since been successfully used in design of numerous approximation algorithms. There is no better approximation guarantee known for congestion minimization in directed or undirected graphs

Congestion minimization is NP-hard even when the optimal congestion value is 1, that is, even when all pairs can be connected in an edge-disjoint manner. Thus no algorithm can approximate congestion better than a factor of two unless P = NP. Until a few years ago, nothing stronger was known on the hardness front. A major progress was made by Chuzhoy and Naor [12] who showed that directed congestion minimization is $\Omega(\log \log n)$-hard to approximate, unless NP is contained in $\mathsf{DTIME}(n^{O(\log \log \log n)})$. The reduction of [12] strongly relies on the directedness of the instances, and APX-hardness continued to be the strongest inapproximability result known for the undirected setting. A big breakthrough was made by Andrews and Zhang [2, 3], who, building on the ideas of Andrews [1], introduced a beautiful new technique for obtaining hardness results for undirected routing problems. In particular, they showed that congestion minimization in undirected graphs is $\Omega(\log \log n)$-hard to approximate. Continuing this thread of research, Andrews and Zhang [4] very recently introduced another elegant idea of encoding paths using labeling schemes, and showed that directed congestion minimization is $\Omega(\log^{1-\epsilon} n)$-hard to approximate for any constant $\epsilon > 0$.

1.1 Our Results

In this paper, building on the framework of [4], we show that directed congestion minimization is $\Omega(\log n / \log \log n)$-hard to approximate unless NP is contained in ZPTIME($n^{\log \log n}$).

We also study the edge disjoint paths problem (EDP), where the input is the same as in congestion minimization, and the goal is to route the maximum number of source-sink pairs on edge-disjoint paths. Directed EDP is known to be $\Omega(n^{\frac{1}{2}-\epsilon})$-hard to approximate [13] for any $\epsilon > 0$. A natural relaxation of the the directed EDP problem is the EDP with congestion problem (EDPwC), where the input additionally includes an integer c, and the goal is to route as many pairs as possible such that congestion on each edge is less than c. The performance of algorithms for EDPwC is compared with the optimal EDP solution of congestion 1. We show that EDPwC is $n^{\Omega(1/c)}$-hard to approximate unless NP is contained in ZPTIME($n^{\text{polylog}(n)}$). This hardness result holds for congestion values c that lie anywhere between 2 and $\delta \log n / \log \log n$, for some fixed constant δ. In particular, when congestion parameter c is in the range $\Theta(1)$ and $\log^{\Theta(1)} n$, we obtain an $n^{1/(3+\epsilon)c}$-hardness for any $\epsilon > 0$. In contrast, it is known that randomized rounding technique can be used to obtain an $O(cn^{1/(c-1)})$-approximation [8, 9, 19]. This result is also the best known approximation for the directed All-or-Nothing Flow problem with congestion (ANFwC), a relaxation of the EDPwC where routing a pair requires only establishing a unit flow from source to sink (as opposed to a single path). In the EDPwC instances created in our hardness reductions, each source-sink pair has a unique path along which it can be connected. As a result, all our results also hold for ANFwC. This is in a sharp contrast to the undirected setting where it was recently shown that All-or-Nothing flow can be approximated to within a factor of $O(\log^2 n)$ without congestion [10, 11]. We note that all our instances are directed acyclic graphs, and our results also extend to the vertex versions of congestion minimization and EDPwC, where congestion is measured on graph vertices and not edges.

The following theorem summarizes our results, more precisely outlined in Theorems 5.4, 6.4 and 7.1, in this paper.

Theorem 1.1 *Congestion minimization is $\Omega(\log n / \log \log n)$-hard to approximate on directed acyclic graphs unless NP $\subseteq$ ZPTIME($n^{\log \log(n)}$). EDPwC and ANFwC are $n^{\Omega(1/c)}$-hard to approximate even on unique paths directed acyclic graphs for $2 \leq c \leq (\delta \log N)/(\log \log N)$, unless NP $\subseteq$ ZPTIME($n^{\text{polylog}(n)}$), where $\delta > 0$ is some universal constant.*

We would like to note that Guruswami and Talwar [14] recently announced that they have proved that directed EDPwC is $n^{\Omega(1/c)}$-hard to approximate for any *constant c*. Our $n^{\Omega(1/c)}$-hardness results for EDPwC for all c upto $\Theta(\log n / \log \log n)$ was obtained subsequent to this announcement.

1.2 Our Techniques

A common theme that runs through the recent hardness proofs for routing problems is the idea of "canonical" and "non-canonical" paths. Roughly speaking, each source-sink pair is assigned a set of special paths in the graph, called its *canonical paths*, and any other path connecting the pair is a non-canonical path. The canonical paths are used to encode some desirable property. For example, one can encode the independent set problem as an instance of EDP: Given an independent set instance $G = (V, E)$, define a source-sink pair for every vertex $v \in V$, and define a canonical path for each source-sink pair. For every edge $e = (u, v) \in E$, there is an edge representing it in the EDP instance, and the two canonical paths representing the endpoints u and v of e traverse this edge in the EDP instance. Thus, if we restrict EDP solutions to canonical paths only, it is easy to see that this problem is at least as hard to approximate as the independent set problem. The main obstacle to proving hardness of approximation for this type of routing problems is therefore the existence of non-canonical paths. In particular, in the above reduction from independent set to EDP, a solution to the EDP instance may not route the source-sink pairs on their canonical paths and choose non-canonical paths instead. Therefore, the correspondence between EDP solutions and independent sets in the original graph does not hold.

As a way to overcome this problem in the context of directed graphs, Andrews and Zhang [4] proposed to use labeling schemes. Roughly speaking, a labeling scheme is a mechanism to enforce that the only paths connecting source-sink pairs are the canonical paths. The hardness result thus obtained depends on the efficiency of the labeling scheme. Andrews and Zhang [4] designed a labeling scheme that gives $\Omega\left(\log^{1-\epsilon}\right)$-hardness for directed congestion minimization, for any $\epsilon > 0$. It is also possible to adapt their ideas in a straightforward manner to show that EDPwC is $\Omega\left(2^{\log^{1/2-\epsilon} n}\right)$-hard to approximate. A key idea underlying all our results is a more efficient labeling scheme. This labeling scheme combined with the framework of [4] suffices for our hardness result for EDPwC when the congestion is at least $\log^{\epsilon} n$ for some $\epsilon > 0$. For smaller values of congestion, we use a somewhat different approach based on a reduction from the independent set problem. The idea is to establish and utilize a simple property that if an n-vertex graph does not have an independent set of size $n^{\Omega(1/c^6)}$, then it contains $n^{\Omega(c)}$ cliques of size c. Our reduction translates cliques of size c into edges with congestion c in the EDPwC instance.

Organization

We start with some preliminaries in Section 2. In Section 3, we describe our labeling scheme. As a simple application of the new labeling scheme, we present in Section 4 an elementary construction to establish the integrality gap of $n^{\Omega(1/c)}$ for the multicommodity flow relaxation of EDPwC. In Section 5, we show hardness of EDPwC for congestion values up to $\log^{\lambda} n$ for *some* $\lambda > 0$. We complement this hardness result in Section 6 with another reduction that obtains similar bounds when the congestion allowed ranges from $\log^{\lambda} n$ for *any* $\lambda > 0$ to $\Theta(\log n / \log\log n)$. Finally, in Section 7, we establish the hardness of directed congestion minimization.

2 Preliminaries

2.1 Problem Statement

Definition: (Congestion Minimization) *Given a graph $G(V, E)$ and a collection of source-sink pairs $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, find the minimum integer c, such that there exists a routing that connects each source s_i to its sink t_i by a path, with at most c paths going through any edge.*

Definition: (EDP) *Given a graph $G(V, E)$ and a collection of source-sink pairs $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, route as many pairs as possible on edge-disjoint paths.*

Definition: (EDPwC) *Given a graph $G(V, E)$, a collection of source-sink pairs $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, and an integer c, route as many pairs as possible such that* less than *c paths go through any edge.* The performance of algorithms for EDPwC is compared to optimal solutions with congestion 1.

We note that all three problems can be defined more generally where each edge e in the graph has a capacity $u(e)$ and each pair (s_i, t_i) has a demand d_i. A solution with congestion c allows up to $c \cdot u(e)$ demand to be routed through each edge e. However, since all our hardness results hold for the restricted version above, we will work throughout this paper with the simple versions defined above. Moreover, we will assume that all source-sink pairs are distinct; instances created by our reductions will satisfy this property.

We will use N to denote the instance size throughout. We will say an instance of EDPwC is a *unique paths* instance if every source-sink pair in the instance has a unique path that connects the source to the sink.

2.2 A Multicommodity Flow Relaxation

Given an instance of the Congestion Minimization or EDPwC, let $\mathcal{P}_i$ denote the set of paths joining s_i and t_i in G and let $\mathcal{P} = \cup_i \mathcal{P}_i$. We define for each path $P \in \mathcal{P}$, a variable $f(P)$ which is the amount of flow sent on P. We also let x_i denote the total flow sent on paths for pair i. Then the multicommodity flow relaxation for Congestion Minimization is as follows:

$$\begin{aligned} &\min\ c \quad \text{s.t} \\ 1 - \sum_{P \in \mathcal{P}_i} f(P) &= 0 \qquad 1 \le i \le k \\ \sum_{P: e \in P} f(P) &\le c \quad e \in E \\ f(P) &\in [0,1] \quad 1 \le i \le k, P \in \mathcal{P}. \end{aligned}$$

While the multicommodity flow relaxation for EDP is as follows:

$$\begin{aligned} &\max \sum_{i=1}^{k} x_i \quad \text{s.t} \\ x_i - \sum_{P \in \mathcal{P}_i} f(P) &= 0 \qquad 1 \le i \le k \\ \sum_{P: e \in P} f(P) &\le 1 \quad e \in E \\ x_i, f(P) &\in [0,1] \quad 1 \le i \le k, P \in \mathcal{P}. \end{aligned}$$

We use the same multicommodity flow relaxation for EDPwC, and we are interested in comparing fractional solutions with no congestion to integral solutions with congestion less than c.

Even though as stated above, these formulations are exponential-size, there is a standard equivalent flow-based encoding of these linear programs that is polynomial-size.

3 Labeling Scheme

One of the primary ingredients used in our results is a *labeling scheme* that takes two integer parameters m and Z, and is denoted by $\mathcal{L}(m, Z)$. The labeling scheme $\mathcal{L}(m, Z)$ consists of a set $U = U(m)$ of m *increment vectors*, and a set $Y = Y(m, Z)$ of *labels*. Let $U = \{u^1, \ldots, u^m\}$ be the set of the increment vectors. Each increment vector u^i, $1 \le i \le m$, has two coordinates, and it is defined to be $u^i = (i, i^2)$. For any integer p, we denote by $[p]$ the set $\{0, 1, 2, ..., p-1\}$. Then the set Y of labels is defined to be all two-dimensional vectors in $[2m^2Z] \times [2m^2Z]$. Thus $|Y| = 4m^4Z^2$. We define the addition operation on the set Y of labels to be the usual vector addition modulo $2m^2Z$. Since $U \subseteq Y$, this also defines the addition of increment vectors and labels.

The labeling scheme can be naturally associated with an instance of EDP, as follows (we present it here only to provide intuition). It is a layered graph that has Z layers. For each layer $z : 1 \le z \le Z$

and for each label $y \in Y$, there is a vertex $v(y,z)$. The edges in the graph connect only pairs of vertices in consecutive layers, and they are directed from lower to higher indexed layers. Each edge belongs to one of the m types $1,\ldots,m$. There is a type-i edge between $v(y,z)$ and $v(y',z+1)$, iff $y' = y + u^i$. Thus for each vertex at layers 1,...,Z-1 there is exactly one edge of each type leaving it, and for each vertex at layers 2,...,Z there is exactly one edge of each type entering it. Fix some $y \in Y$, and consider the vertex $v(y,1)$ in the first layer. Let $P_i(y)$ be the unique path that starts at vertex $v(y,1)$, finishes at some vertex of the last layer, and only uses the edges of type i. Let $s(y,i)$ and $t(y,i)$ be the endpoints of this path. Then $s(y,i)$ and $t(y,i)$ is a source-sink pair in the constructed graph. The main requirement from the labeling scheme is that for each $y \in Y$, $i : 1 \leq i \leq m$, $P_i(y)$ is the only path connecting $s(y,i)$ to $t(y,i)$. We will show that our construction achieves this property, while being efficient in the number of labels: in the construction of Andrews and Zhang, the size of the set of labels is roughly $Z^{\log m}$, while in our construction it is $|Y| = O(m^4 Z^2)$.

We need the following claim.

Claim 3.1 *Let $k \leq Z+1$ be any positive integer, and let $u^{i_1},\ldots,u^{i_k},u^j \in U$ be such that not all the vectors $u^{i_1},\ldots,u^{i_k}$ are equal to u^j. Then $ku^j \neq u^{i_1} + u^{i_2} + \cdots + u^{i_k}$.*

Proof: Assume otherwise. In this proof we work with standard vector addition (and not addition modulo $m^2 Z$). Then $u^j = \frac{u^{i_1}+u^{i_2}+\cdots+u^{i_k}}{k}$. Therefore, u^j is a convex combination of the points $u^{i_1},\ldots,u^{i_k}$. Since the curve (x,x^2) is strictly convex, this is only possible if $u^{i_1} = \cdots = u^{i_k} = u^j$. A contradiction. □

We will use the following corollary of Claim 3.1:

Corollary 1 *Let $y, y' \in Y$ be a pair of labels such that there is some $j \in [m]$ for which $y + Zu^j = y'$. Let $u^{i_1},\ldots,u^{i_Z}$ be any collection of Z increment vectors from U. Then $y + \sum_{p=1}^{Z} u^{i_p} = y'$ iff $u^{i_1} = u^{i_2} = \cdots = u^{i_Z} = u^j$.*

Proof: Assume otherwise. Notice that in the increment vectors, all coordinate values are between 1 and m^2. Thus the value of every coordinate in the vector $\sum_{p=1}^{Z} u^{i_p}$ is between Z and $m^2 Z$. The same is true for coordinates of vector Zv^j. Therefore, since the addition is performed modulo $2m^2 Z$, if $y + \sum_{p=1}^{Z} u^{i_p} = y'$ and $y + Zu^j = y'$, then it must be the case that $\sum_{p=1}^{Z} u^{i_p} = Zu^j$. From Claim 3.1, this is only possible if $u^{i_1} = u^{i_2} = \cdots = u^{i_Z} = u^j$. □

Notice that the above corollary shows that $\mathcal{L}(m,Z)$ is indeed a proper labeling scheme, i.e., for each $y \in Y$ and for each $i : 1 \leq i \leq m$, there is a unique path $P_i(y)$ connecting $s_i(y)$ to $t_i(y)$.

4 Integrality Gap

We present an elementary construction that shows that the integrality gap of the multicommodity flow relaxation is $\Omega\left(\frac{N^{\frac{1}{3c+11}}}{c}\right)$ for directed EDP with congestion c, where c can be any integer between 2 and $\delta \log N / \log\log N$, for some fixed constant δ. We will use a labeling scheme $\mathcal{L}(m,Z) = (U,Y)$ with parameters m and Z to be specified later. Let $\{u^i\}_{i=1}^m$ be the increment vectors in U.

Our instance is a layered graph with Z layers and $m|Y|$ source-sink pairs. For each $y \in Y$, for each $i \in [m]$, there is a source-sink pair $s(y,i)$-$t(y,i)$, and we will later define a corresponding canonical path $P(y,i)$. For each layer $z : 1 \leq z \leq Z$, for each label $y \in Y$, we have a set $E(y,z)$ of m/c *special edges*, whose endpoints are disjoint.

A canonical path $P(y,i)$ starts at source $s(y,i)$ and then traverses, for each $z : 1 \leq z \leq Z$ one special edge at layer z. After traversing a special edge at layer Z, it finishes at sink $t(y,i)$. Thus, in order to define a canonical path $P(y,i)$ we need to specify, for each layer z, what is the special edge at layer z that is being traversed by $P(y,i)$. We will then add additional edges to the graph, called the *non-special* edges, which are needed to realize the canonical paths.

It now only remains to assign to each path $P(y,i)$ one special edge at each layer $z : 1 \leq z \leq Z$. Fix some $y \in Y$, $i \in [m]$. We first define, for each $z \in Z$, the label of $P(y,i)$ at layer z to be $y + zu^i$.

Fix some layer $z \in Z$ and some label $y \in Y$. Let $\mathcal{P}_z(y)$ be the set of all the canonical paths $P(y',i)$ such that the label of $P(y',i)$ for layer z is y. Notice that for each $i : 1 \leq i \leq m$, there is exactly one such path $P(y',i)$, and thus $|\mathcal{P}_z(y)| = m$. We randomly partition the set $\mathcal{P}_z(y)$ into sets of size c each. We will refer to each set as a c-tuple. Each c-tuple is assigned one edge in the set $E(y,z)$, and all the canonical paths in this c-tuple go through this edge at layer z.

Finally, we add non-special edges needed to realize the canonical paths. No parallel edges are added. This completes the construction description.

Construction Size

The size of the construction is $N \leq O(mZ|Y|) = O(mZ \cdot 4m^4Z^2) = O(m^5Z^3)$.

Fractional Solution

In the fractional solution, we can route $1/c$-fraction of flow on each canonical path, causing congestion 1. Thus the fractional solution routes at least $m|Y|/c$ units of flow.

Integral Solution

Consider now any integral solution with congestion strictly less than c. We will show that it routes at most $m|Y|/cg$ flow where g will be the integrality gap.

Lemma 4.1 *For each source-sink pair in our construction, there is only one path connecting the source to the sink (the canonical path).*

Proof: Consider any source-sink pair $s(y,i)$-$t(y,i)$. Assume for contradiction that there is a non-canonical path P' connecting $s(y,i)$ to $t(y,i)$. Let $y = y_0, y_1, y_2, \ldots, y_Z$ be the sequence of labels of the source and the special edges appearing on path P', and let $u^{j_1}, \ldots, u^{j_Z}$ be the collection of increment vectors used along this path, such that for each $k : 1 \leq k \leq Z$, $y_k = y_{k-1} + u^{j_k}$. Since P' is a non-canonical path, and since for each special edge in the graph, there is at most one edge corresponding to each increment vector leaving it, it must be the case that at least one of the increment vectors u^{j_p}, $1 \leq p \leq Z$ differs from u^i. But then, from Corollary 1, it is impossible that P' reaches $t(y,i)$. $\square$

Lemma 4.2 *Suppose $Z \geq (8c)^{c+3} g^{c+2}$ and $g \leq m/8c^2$. Then if more than $m|Y|/cg$ pairs are routed, then at least one edge has congestion c, with high probability.*

Proof: Let $\mathcal{S}$ be any collection of at least $m|Y|/cg$ canonical paths. We say a label y at a layer z is *good* if there are at least $m/2cg$ paths $P(y', i) \in \mathcal{S}$, which have label y at layer z. At least $|Y|/2cg$ labels must be good at each layer z. Otherwise, total number of paths contained in $\mathcal{S}$ is *strictly less than*

$$\left(\frac{|Y|}{2cg}\right) m + |Y| \left(\frac{m}{2cg}\right) = \left(\frac{m|Y|}{cg}\right).$$

A contradiction! Thus there must be at least $(|Y|)/2cg$ good labels at each layer.

Consider a good label y at a layer z. We say that a bad event $B(y, z)$ happens iff there is no edge $e \in E(y, z)$ with congestion c. We now bound the probability of $B(y, z)$. We consider the first $m/4c^2 g$ edges $e_1, \ldots, e_{m/4c^2 g}$ in $E(y, z)$, and assume that each edge chooses a c-tuple of paths in this order. For each $j : 1 \leq j \leq m/4c^2 g$, no matter what is the configuration of edges $e_1, \ldots, e_{j-1}$, the probability that edge e_j has chosen all its c paths from $\mathcal{S}$ is at least:

$$\begin{aligned}
\frac{\binom{\frac{m}{4cg}}{c}}{\binom{m}{c}} &= \frac{\frac{m}{4cg} \cdot \left(\frac{m}{4cg} - 1\right) \cdots \left(\frac{m}{4cg} - c + 1\right)}{m \cdot (m-1) \cdots (m-c+1)} \\
&\geq \left(\frac{\frac{m}{4cg} - c + 1}{m}\right)^c \geq \left(\frac{1}{8cg}\right)^c
\end{aligned}$$

Thus

$$\Pr[B(y, z) \text{ occurs}] \leq \left(1 - \frac{1}{(8cg)^c}\right)^{m/4c^2 g} \leq e^{-\frac{m}{(4c^2 g)(8cg)^c}} \leq e^{-\frac{m}{(8c)^{c+2} g^{c+1}}}$$

Let B be the event defined by the intersection of all events $B(y, z)$ for all pairs (y, z) such that the label y is good at layer z. Then using the assumption $Z \geq (8c)^{c+3} g^{c+2}$, we get

$$\Pr[B \text{ occurs}] \leq e^{-\frac{m|Y|Z}{2cg(8c)^{c+2} g^{c+1}}} \leq e^{-m|Y|}.$$

On the other hand, the number of possible solutions $\mathcal{S}$ of size $m|Y|/cg$ can be bounded by $2^{m|Y|} = o(e^{m|Y|})$. Using union bounds, we conclude that with high probability there is an edge with congestion c. □

Recall that the construction size is $N \leq O(m^5 Z^3)$. Substituting $Z = (8c)^{c+3} g^{c+2}$ and $g = m/8c^2$, we get that

$$N \leq O(m^5 (8c)^{3c+9} g^{3c+6}) \leq O((8c^2)^5 (8c)^{3c+9} g^{3c+11}) \leq O((8c)^{3c+20} g^{3c+11}).$$

Thus $g = \Omega(\frac{N^{\frac{1}{3c+11}}}{c})$. Hence we obtain the following theorem.

Theorem 4.1 *There exists a constant $\delta > 0$, such that the integrality gap of the multicommodity flow relaxation for directed EDPwC is $\Omega(\frac{N^{\frac{1}{3c+11}}}{c})$ for any congestion $c \leq (\delta \log N)/(\log \log N)$.*

5 Hardness of Directed EDP for Small Congestion Values

In this section we prove that directed EDP with congestion c is $N^{\Omega(1/c)}$-hard to approximate for any c between 2 and $\log^{\lambda} N$, where λ is some fixed constant, $0 < \lambda < 1$. We first show hardness for any constant c, and later show how it can be extended to higher values of c.

5.1 The Construction

We perform a reduction from the independent set problem. Given a graph $G = (V, E)$, a subset $S \subseteq V$ of vertices is called *independent set* iff the subgraph induced by S does not contain any edges of E. Our starting point is the following result of Håstad [15].

Theorem 5.1 [15] *For any $\epsilon > 0$, no polynomial-time algorithm can distinguish between n-vertex graphs that have an independent set of size at least $n^{1-\epsilon}$* (the YES-INSTANCE) *and graphs that have no independent sets of size greater than n^{ϵ}* (the NO-INSTANCE)*, unless NP is contained in ZPP.*

Given an instance G of the independent set problem above, we will construct a unique paths instance I of directed EDP such that if G is a YES-INSTANCE, then we can route an $\Omega(1/n^{\epsilon})$-fraction of the pairs with congestion 1. On the other hand, if G is a NO-INSTANCE, we will show that even when a congestion of $c-1$ is allowed, no more than $O(1/\sqrt{n})$-fraction of pairs can be routed. We assume that $\epsilon < \frac{1}{2c^6}$.

We will use the labeling scheme $\mathcal{L}(n, Z) = (U, Y)$, where n is the number of vertices in the independent set instance, and Z will be specified later. As in the integrality gap construction above, we build a layered graph with Z layers. For each layer $z : 1 \leq z \leq Z$, and for each label $y \in Y$, we have n/c blobs $B(y, z, i)$ of special edges, where $1 \leq i \leq n/c$. Each blob will either contain one special edge, or c special edges. This is determined as follows. For each label $y \in Y$, for each layer $z \in Z$, we choose a random partition $\mathcal{P}_1(y, z), \ldots, \mathcal{P}_{n/c}(y, z)$ of the vertices in $V(G)$ into c-tuples. For each $i : 1 \leq i \leq n/c$, if $\mathcal{P}_i(y, z)$ is a clique in G, then the blob $B(y, z, i)$ is called a type-1 blob, and it has just one special edge. If $\mathcal{P}_i(y, z)$ is not a clique, then blob $B(y, z, i)$ is called a type-2 blob, and it has c special edges.

For each vertex $v \in V(G)$, for each label $y \in Y$, there is a source-sink pair $s(y, v)$-$t(y, v)$, and a corresponding canonical path $P(y, v)$. This canonical path starts at $s(y, v)$, and then traverses, for each $z : 1 \leq z \leq Z$, one special edge at level z, in this order. After visiting a special edge from layer Z, the path ends at $t(y, v)$. Next we specify what are exactly the special edges that each canonical path visits at each layer. After this we add all the non-special edges that are needed to realize the canonical paths.

We first define, for each layer $z : 1 \leq z \leq Z$, for each $v \in m$ and for each $y \in Y$, the label of $P(y, v)$ at layer z to be $y + zu^v$. At each layer $z : 1 \leq z \leq Z$, the path $P(y, v)$ goes through the blob $B(y', z, i)$, where $y' = y + zu^v$, and i is such that v belongs to $\mathcal{P}_i(y, z)$. If blob $B(y', z, i)$ is a type-1 blob, then $P(y, v)$ traverses the unique special edge belonging to this blob. If blob $B(y', z, i)$ is a type-2 blob, then path $P(y, v)$ traverses the unique edge in blob $B(y', z, i)$ which is assigned to it (we note that since there are exactly c canonical paths visiting any blob, we can assign a unique special edge to each path visiting a type-2 blob). We add all the non-special edges to the graph that are needed to realize the canonical paths. No parallel edges are added.

Construction Size

Let N denote the size of the instance I constructed above. Then $N = O(Z|Y|n) = O(Z^3 n^5)$.

Yes-Instance

If G is a YES-INSTANCE, then it has an independent set S of size at least $n^{1-\epsilon}$. For each $v \in S$ and $y \in Y$, we route the source-sink pair $s(y,v)$-$t(y,v)$ on its canonical path $P(y,v)$. This gives a set of $|Y||S| \geq |Y|n^{1-\epsilon}$ edge disjoint paths.

No-Instance

We show that any subset $\mathcal{M}$ of $4|Y|\sqrt{n}$ source-sink paths causes congestion c, with high probability.

Lemma 5.1 *There are no non-canonical source-sink paths in the graph.*

Proof: Follows from the labeling scheme construction and Corollary 1. □

Let $T(\alpha, c)$ denote the minimum number of c-cliques in any graph on α vertices, which does not contain an independent set of size s. We need the following lemma.

Lemma 5.2 *Let $c \geq 2$ be a positive integer. Then for any $\alpha > (4s)^c$, $T(\alpha, c) \geq \frac{\alpha^c}{(2c)^c (4s)^{c^3}}$.*

Proof: We will use the following simple fact: if a graph has at least k vertices with degree at most d, then it contains an independent set of size at least $k/(d+1)$. This follows easily by restricting attention to vertices of degree at most d and choosing a maximal independent set from them.

We will prove the lemma by induction on c. Base case is $c = 2$. Assume that $\alpha > (4s)^2$. Let H be any graph with α vertices such that the average degree is d. Then H contains an independent set of size at least $\alpha/(4d+2)$ since at least half the vertices have degree at most $2d$. Therefore, $\alpha/(4d+2) < s$ and $d > \alpha/5s$. Thus the number of edges (cliques of size 2) in the graph is $\alpha d/2 > \alpha^2/10s$.

For the induction step, observe that at least $\alpha/2$ vertices in H must have degree at least $d = \alpha/2s - 1 \geq \alpha/4s$: otherwise, we can find an independent set of size s in H.

Let v be a vertex in H of degree at least d. Consider the neighborhood of v. Since $\alpha > (4s)^c$, the number of neighbors of v is at least $d \geq \alpha/4s > (4s)^{c-1}$. Therefore, by induction hypothesis, the neighborhood of v contains at least $T(d, c-1)$ cliques of size $c-1$. Each such clique is a c-clique in H. Counting these cliques for all vertices in H with degree at least d and compensating for the fact that a c-clique may get counted up to c times, we get

$$T(\alpha, c) \geq \frac{\alpha}{2c} T(\alpha/4s, c-1).$$

Iterating, we get

$$
\begin{aligned}
T(\alpha, c) &\geq \frac{\alpha}{2c} \cdot \frac{\alpha/4s}{2(c-1)} \cdot \frac{\alpha/(4s)^2}{2(c-2)} \cdots \frac{\alpha/(4s)^{c-3}}{2(3)} T(\alpha/(4s)^{c-2}, 2) \\
&\geq \frac{\alpha^{c-2}}{(2c)^{c-2}(4s)^{c^2/2}} T(\alpha/(4s)^{c-2}, 2) \\
&\geq \frac{\alpha^c}{(2c)^c (4s)^{c^3}}.
\end{aligned}
$$

□

In what follows, let $s = n^\epsilon$ (recall that $\epsilon < 1/c^6$).

Corollary 2 *Any graph H on $\alpha = \sqrt{n}$ vertices that does not contain an independent set of size s has at least $n^{\frac{c}{2} - \frac{1}{c^2}}/c^c$ distinct cliques of size c.*

Proof: Follows immediately from the above Lemma, using the facts that $\epsilon < 1/c^6$ and $c < \log^\lambda N$ for some small $\lambda < 1/4$. □

Assume now that there is a solution that routes a set $\mathcal{M}$ of $4|Y|\sqrt{n}$ canonical paths. We show that with high probability, there is at least one edge with congestion c.

A label y is called a *good label* for layer z, iff the number of canonical paths $P(y, v) \in \mathcal{M}$ whose label at layer z is y is at least $2\sqrt{n}$.

Claim 5.1 *For each layer $z : 1 \leq z \leq Z$, the fraction of labels y which are good for layer z is at least $2/\sqrt{n}$.*

Proof: Assume otherwise. The number of paths of $\mathcal{M}$ that go through good labels at layer z is at most $2|Y|n/\sqrt{n} = 2|Y|\sqrt{n}$. The number of paths of $\mathcal{M}$ that go through labels which are not good is less than $|Y| \cdot 2\sqrt{n}$. Thus, in total we get that $\mathcal{M}$ contains less than $4|Y|\sqrt{n}$ paths. □

Let y be a good label for layer z. We say that the bad event $B(y, z)$ happens iff no special edge at any blob belonging to (y, z) has congestion c.

Claim 5.2 *The probability of $B(y, z)$ happening is at most* $\exp\left(-\frac{1}{c^{c+1} n^{\frac{c}{2} - \frac{1}{2} + \frac{1}{c^2}}}\right)$.

Proof: Let S be the subset of vertices in G, corresponding to paths in $\mathcal{M}$ whose label at layer z is y, $|S| \geq 2\sqrt{n}$. We consider the random choices made by the construction as follows: the n/c c-tuples of vertices are chosen one after another. We focus on the choice of the first $\sqrt{n}/c$ tuples. These choices are not independent. However, when the ith choice is made, if S_i denotes all the vertices chosen at steps $1, \ldots, i-1$, then $S \setminus S_i$ still contains at least $\sqrt{n}$ vertices, and thus the graph induced by $S \setminus S_i$ contains at least $n^{\frac{c}{2} - \frac{1}{c^2}}/c^c$ cliques of size c. The probability of choosing such a clique at step i is at least $\frac{n^{\frac{c}{2} - \frac{1}{c^2}}}{c^c n^c} = \frac{1}{c^c n^{\frac{c}{2} + \frac{1}{c^2}}}$. Thus the probability of $B(y, z)$ happening is at most:

$$\left(1-\frac{1}{c^c n^{\frac{c}{2}+\frac{1}{c^2}}}\right)^{\sqrt{n}/c} \leq \exp\left(-\frac{1}{c^{c+1} n^{\frac{c}{2}-\frac{1}{2}+\frac{1}{c^2}}}\right)$$

$\square$

Assume now that event $B(y,z)$ happened to all pairs (y,z) where y is good for z. The probability of this is at most:

$$\exp\left(-\frac{2|Y|Z}{c^{c+1} n^{\frac{c}{2}+\frac{1}{c^2}}}\right)$$

If we set $Z = c^{c+1} n^{\frac{c}{2}+1+\frac{1}{c^2}}$, then this probability is bounded by $e^{|Y|n}$. The total number of possible solutions is at most $2^{|Y|n} = o\left(e^{|Y|n}\right)$. Therefore, using the union bound, with high probability there is a label y and a layer z for which $B(y,z)$ does not happen. This means that at least one edge has congestion c, with high probability.

Notice that the gap that we obtain is $n^{\frac{1}{2}-\epsilon}$, while the construction size is:

$$N = O(Z^3 n^5) = O(c^{3c+3} n^{\frac{3c}{2}+9})$$

The gap is thus $\Omega\left(N^{\frac{1}{3c+19}}/c\right)$, since $\epsilon < 1/c^6$. Note that for $c \leq \log^{1/4} N$, $\left(N^{\frac{1}{3c+19}}/c\right)$ is $\Omega\left(N^{\frac{1}{3c+20}}\right)$ We thus get the following theorem.

Theorem 5.2 *For any fixed positive constant integer $c \geq 2$, directed EDP with congestion c is hard to approximate within a factor of $\Omega\left(N^{\frac{1}{3c+20}}\right)$ unless NP $\subseteq$ ZPP.*

If we slightly relax the complexity assumptions of Theorem 5.1, we can use the theorem below due to Khot [16]:

Theorem 5.3 [16] *There exists a constant $0 < \gamma < 1$ such that no polynomial-time algorithm can distinguish between n-vertex graphs that have an independent set of size at least $\frac{n}{2^{(\log n)^{1-\gamma}}}$* (the Yes-Instance) *and graphs that have no independent sets of size greater than $2^{(\log n)^{1-\gamma}}$* (the No-Instance)*, unless NP $\subseteq$ ZPTIME($n^{\text{polylog}(n)}$).*

Theorem 5.4 *There exists a constant $0 < \lambda < 1/4$ such that for any $c \leq (\log N)^{\lambda}$, directed EDP with congestion c is hard to approximate within a factor of $\Omega\left(N^{\frac{1}{3c+20}}\right)$ unless NP $\subseteq$ ZPTIME($n^{\text{polylog}(n)}$).*

6 Hardness of Directed EDP for High Congestion Values

In this section we show that EDPwC is $N^{\Omega(1/c)}$-hard to approximate for $\log^{\lambda} N \leq c \leq \delta \log N/\log\log N$ unless NP has randomized quasi-polynomial time algorithms, where δ is some fixed constant, while λ is any constant $0 < \lambda < 1$.

Our hardness proof builds on the paper of Andrews and Zhang [4], in which they prove $\Omega\left(\log^{1-\epsilon}\right)$-hardness of directed congestion minimization. It is not hard to see that reasoning similar to [4] can be used to prove that EDPwC is $\Omega\left(2^{\log^{1/2-\epsilon} n}\right)$-hard to approximate. The main obstacle to obtaining better hardness of approximation result for both EDPwC and congestion minimization is the large size of the graph constructed in [4]. Specifically, to obtain better hardness of approximation result by using the same approach, one needs to reduce the size of the set of labels. We achieve this by using the labeling scheme described in Section 3.

6.1 Starting Point

We perform a reduction from the 3SAT(5) problem, which is defined as follows. The input is a CNF formula with n variables and $5n/3$ clauses. Each clause contains exactly 3 literals, and each variable appears in exactly 5 different clauses. The goal is to find an assignment maximizing the number of satisfied clauses. The following theorem is one of the several alternative statements of the PCP theorem [7, 6].

Theorem 6.1 *There is some constant $\epsilon > 0$, such that it is NP-hard to distinguish between a satisfiable 3SAT(5) formula, and a formula where no assignment can satisfy more than a fraction $(1-\epsilon)$ of the clauses.*

If a 3SAT(5) formula φ is satisfiable, we call it a YES-INSTANCE. If any assignment satisfies at most a fraction $(1-\epsilon)$ of the clauses, then it is called a NO-INSTANCE.

We use the symmetric version of Raz verifier, with ℓ repetitions, where ℓ will be specified later. This is an interactive proof system, in which two provers try to convince the verifier that the input 3SAT(5) formula φ is satisfiable. Given input 3SAT(5) formula φ, the verifier chooses ℓ random clauses from φ, and for each clause, one distinguished variable is chosen. The first prover receives the indices of the first $\ell/2$ clauses and the last $\ell/2$ distinguished variables, while the second prover receives the indices of the first $\ell/2$ distinguished variables and the last $\ell/2$ clauses. The provers respond with an assignment to all the variables appearing in their queries, both as distinguished variables and as parts of clauses. The assignment returned by each prover must satisfy all the clauses that appear in its query, and this is checked by the verifier. Additionally, the verifier checks that the answers of the two provers are consistent, that is, for each one of the ℓ clauses, the assignments that both provers return to its distinguished variable, are identical.

The following theorem follows from Theorem 6.1 combined with the Raz Parallel Repetition Theorem [18].

Theorem 6.2 *There is a constant $\alpha > 0$, such that:*

- *If φ is a* YES-INSTANCE, *then there is a strategy of the provers that makes the verifier accept always.*
- *If φ is a* NO-INSTANCE, *then no matter what the strategy of the provers is, the verifier accepts with probability at most $2^{-\alpha\ell}$.*

Let Q_1, Q_2 be the sets of all the queries to the first and the second provers, respectively, $|Q_1| = |Q_2| = (5n/3)^{\ell/2} \cdot n^{\ell/2} = n^{\ell} \cdot (5/3)^{\ell/2}$. Let R be the set of all the random strings of the verifier, $|R| = (5n)^{\ell}$. For a random string $r \in R$, let $q_1(r), q_2(r)$ be the queries sent to the two provers when the verifier chooses r. Let $A(r)$ be the set of all the assignments to all the variables appearing in the clauses associated with r, where each assignment in $A(r)$ satisfies all these clauses. Notice that $|A(r)| = 7^{\ell}$. For each query $q \in Q_1 \cup Q_2$, let $A(q)$ denote the set of all the possible answers to query q that satisfy all the clauses in this query. Then $|A(q)| = 7^{\ell/2} \cdot 2^{\ell/2} = 14^{\ell/2}$. Notice that for each query $q \in Q_1 \cup Q_2$, the number of random strings in which q participates is exactly $3^{l/2} \cdot 5^{\ell/2} = 15^{\ell/2}$.

We now proceed as follows. We start by defining a proof graph P, after which we use graph P together with the labeling scheme to construct the final graph T.

6.2 Proof Graph P

For each query $q \in Q_1 \cup Q_2$, for each answer $a \in A(q)$, this graph contains a *special edge* $e(q, a)$. The endpoints of all the special edges are disjoint. The source-sink pairs in this graph are the following. For each random string $r \in R$, for each assignment $b \in A(r)$, there is a source-sink pair $s(r, b)$-$t(r, b)$. We now define, for each source-sink pair $s(r, b)$-$t(r, b)$, a *canonical path* $P(r, b)$ connecting it. The canonical path $P(r, b)$ starts at $s(r, b)$, then traverses the special edges $e(q_1, a_1), e(q_2, a_2)$, where $q_1 = q_1(r), q_2 = q_2(r)$, and a_1, a_2 are the unique answers to queries q_1 and q_2, respectively, that are consistent with b. After traversing the above two edges, the path ends at $t(r, b)$. We add to graph P all the edges needed to realize the canonical paths. These edges are called *non-special edges*.

Consider any query-answer pair (q, a), where $q \in Q_1 \cup Q_2$, $a \in A(q)$. We define a set $D(q, a)$ of demands (r, b), such that the canonical path $P(r, b)$ traverses the edge $e(q, a)$. Notice that the number of canonical paths traversing any such edge is at most $15^{l/2} \cdot 4^{l/2} < 8^l$, since query q participates in $15^{l/2}$ random strings, and for any random string $r \in R$ in which q participates, there are at most $4^{l/2}$ assignments $b \in A(r)$ such that a is consistent with b. We denote $I = 8^l$, and thus $|D(q, a)| < I$.

6.3 Transformed Graph T

We use the labeling scheme $\mathcal{L}(m, Z)$ where $m = |R| \cdot 7^{\ell} = n^{O(\ell)}$, and parameter Z will be defined later. Let $U = U(m)$ be the set of all the increment vectors, and we assume that for each $r \in R$, $b \in A(r)$, there is a unique vector $u(r, b) \in U$ associated with it. Let Y denote the set of labels, $|Y| = O(m^4 Z^2)$.

We now define the transformed graph T. This graph has Z layers. Fix some layer $z : 1 \leq z \leq Z$. For each label $y \in Y$, for each query $q \in Q_1 \cup Q_2$, there is a blob $B(q, y, z)$ of I *special edges*. The endpoints of all the special edges are distinct. The set of demands and source-sink pairs is defined as follows. For each label $y \in Y$, for each $r \in R$ and for each assignment $b \in A(r)$, there is a source-sink pair $s(r, b, y)$-$t(r, b, y)$ and a canonical path $P(r, b, y)$.

Consider some demand (r, b, y), and let $z : 1 \leq z \leq Z$ be some layer. The label of (r, b, y) at layer z is defined to be $y + zu(r, b)$.

We now define the canonical path $P(r, b, y)$. This canonical path will start at $s(r, b, y)$, and traverses, for each $z : 1 \leq z \leq Z$, an edge in a blob $B(q_1(r), y', z)$, and an edge in a blob $B(q_2(r), y', z)$, where y' is the label of (r, b, y) at layer z. After traversing an edge corresponding to prover 2 at the last layer, the path ends at $t(r, b, y)$. We now show how the specific special edges traversed by a canonical path in each blob are determined.

Consider now some blob $B(q, y, z)$, for some $q \in Q_1 \cup Q_2, y \in Y, z \in Z$ (assume w.l.o.g. that $q \in Q_1$). Consider all the canonical paths that need to traverse an edge in this blob. These are all the demands (r, b, y') for which $q_1(r) = q$ and the label of (r, b, y') at layer z is y. Notice that for each pair (r, b), such that $q_1(r) = q$ and $b \in A(r)$, there is exactly one canonical path $P(r, b, y')$ that goes through blob $B(q, y, z)$.

Consider an edge $e(q, a)$ in graph P, where $a \in A(q)$. Recall that $D(q, a)$ is the set of all the demands whose canonical paths in P go through this edge, and $|D(q, a)| = I$. We randomly map the paths in $D(q, a)$ to edges in $B(q, y, z)$. We perform this procedure for every edge $e(q, a)$ in P. Finally, if some path $P(r, b)$ of graph P is mapped to an edge e in blob $B(q, y, z)$, then the unique canonical path $P(r, b, y')$ of graph T that needs to traverse this blob will do so via the edge e. We add non-special edges that are needed to realize the canonical paths. We do not add parallel edges to the graph.

Construction Size

The size of our construction is $N = 2|Y||Z||Q|I \leq O(Z^3) \cdot n^{O(\ell)}$. We will choose Z to be $2^{\Theta(\ell c + \ell \log n)}$. Hence our construction size is $2^{\Theta(\ell c + \ell \log n)}$.

Yes-Instance Analysis

In the YES-INSTANCE, we route, for each random string $r \in R$, for each label $y \in Y$, the pair $s(r, b, y)$-$t(r, b, y)$ along its canonical path $P(r, b, y)$, where b is the "correct" assignment to the variables associated with r. Since for each layer z, for each label y, for each random string r, and for each $q \in \{q_1, q_2\}$ there is only one canonical path of the form $P(r, b, y')$ traversing the blob $B(q(r), y, z)$, and since all the random strings that share a query will be choosing identical answers to this query, the routing is edge-disjoint. We denote the cost of the solution by $C_{YI} = |Y||R|$.

No-Instance Analysis

The goal of this section is to show that if more than C_{YI}/g source-sink pairs are routed, then we have an edge with congestion c, with high probability (we will set the value of g later, and it will be our hardness gap).

We start by showing that all the demands are routed on canonical paths only.

Lemma 6.1 *For each source-sink pair in graph T, there is a unique path from the source to the sink (the canonical path).*

Proof: Assume otherwise. Let (r, b, y) be the demand for which there is a non-canonical path P'. Let z be the first layer when path P' differs from the canonical path, and let y^* be the label of $P(r, b, y)$ at layer z. From Corollary 1, it is enough to prove that the path P' uses at least one non-special edge whose corresponding increment vector is different from $u(r, b)$.

Two cases are possible.

Case 1: If path P' first diverges from the canonical path while going from layer z to layer $z + 1$, then this is only possible if path P' uses a non-special edge that belongs to some different demand

$(r', b') \neq (r, b)$ (this is since for each special edge, for each fixed pair (r'', b''), there is at most one demand (r'', b'', y'') that goes through this edge). Thus, when path P' diverges from the canonical path, we add $u(r', b') \neq u(r, b)$ to the label. Following Corollary 1, this path cannot reach the correct sink.

Case 2: If the second case occurs, then path P', while going from a prover-1 blob to a prover-2 blob at layer z, uses an edge that does not belong to the canonical path. Suppose this edge belongs to some other canonical path $P(r', b', y^{**})$. Assume first that $r \neq r'$. Then at the next step path P' is traversing an edge of a blob $B(q', y^*, z)$, where $q' \neq q_2(r)$, and thus no canonical path belonging to pair (r, b) goes through this blob. When leaving this blob, an edge belonging to some other canonical path $P(r'', b'', y'')$ must be used, where $(r'', b'') \neq (r, b)$. Therefore, path P' uses a non-special edge whose increment vector is $u(r'', b'') \neq u(r, b)$, and thus it cannot reach the sink $t(r, b, y)$, from Corollary 1.

Now assume that path P' uses a non-special edge belonging to some canonical path $P(r, b', y')$ where $b' \neq b$. Then at the next step, path P' will traverse a special edge of the blob $B(q_2(r), y^*, z)$. However, this is a different edge than the one traversed by the canonical path $P(r, b, y)$, which also goes through an edge in the same blob (the edge is different since P' diverges from the canonical path at this point, and there are no parallel edges in the graph). But for each pair (r, b), $r \in R$, $b \in A(r)$, there can be only one canonical path of the form $P(r, b, y')$ that goes through any blob, and thus, when leaving this edge, path P' will have to use a non-special edge that belongs to some canonical path $P(r', b'', y'')$, where $(r, b) \neq (r', b'')$. Therefore, as in previous cases, path P' uses a non-special edge whose increment vector is different from $u(r, b)$, which is impossible. □

Theorem 6.3 *Let $g : 2 \leq g \leq 2^{\alpha\ell}/8c^2$, and let $Z = 2^{O(\ell(c+\log n))}$. Let $\mathcal{S}$ be any solution to the above instance that routes more than C_{YI}/g demands. Then with high probability, there is at least one edge with congestion c.*

Proof: Let $\mathcal{S}$ be any solution that routes more than $C_{YI}/g = |Y||R|/g$ demands. Fix some label $y \in Y$ and layer $z : 1 \leq z \leq Z$. We say that a pair (y, z) is *good* if there are at least $|R|/2g$ demands that are routed in $\mathcal{S}$ and whose label at layer z is y. Then the number of good pairs (y, z) is at least $|Y|Z/2g$, from simple counting arguments.

We now concentrate on a good pair (y, z). Fix some query $q \in Q_1 \cup Q_2$ and consider all the demands $(b, r, y') \in \mathcal{S}$ that use the blob $B(q, y, z)$. For each answer $a \in A(q)$, we define a subset $D_{y,z}(q, a) \subseteq D(q, a)$ of all the demands (b, r, y'), such that demand (b, r) uses the edge $e(q, a)$ in P, and demand (b, r, y') belongs to $\mathcal{S}$ and goes through the blob $B(q, y, z)$. We say that the answer a to query q is *heavy* for (y, z) iff $|D_{y,z}(q, a)| \geq \frac{15^{\ell/2}}{14^{\ell/2} \cdot 8g}$. Notice: we need to ensure that $g < \frac{15^{\ell/2}}{8c \cdot 14^{\ell/2}}$. However, we have assumed that $g \leq 2^{\alpha\ell}/8c^2$. Since we can assume w.l.o.g. that α is a very small constant, assuming $g < 2^{\alpha\ell}/8c^2$ is enough to ensure the above bound.

We say that a query q is *heavy* for (y, z) iff it has at least c heavy answers. We now proceed as follows. First, we prove that for each good pair (y, z), at least one query is heavy. Next, we prove that in this case, with high probability, at least one edge has congestion c.

Lemma 6.2 *For each good pair (y, z), there is at least one heavy query q.*

Proof: Assume otherwise. Let (y, z) be any good pair for which there is no heavy query q. Define a set $\mathcal{P}$ of canonical paths in graph P, as follows: $P(b,r) \in \mathcal{P}$ iff the canonical path $P(b,r,y')$ in graph T belongs to $\mathcal{S}$, where y' is the unique label, such that the label of (b,r,y') at layer z is y. Since (y,z) is good, $|\mathcal{P}| \geq |R|/2g$.

The number of demands in $\mathcal{P}$ that do not use heavy answer edges is at most $|Q||A| \cdot \frac{15^{\ell/2}}{14^{\ell/2} \cdot 8g} \leq n^{\ell}\left(\frac{5}{3}\right)^{\ell/2} \cdot 14^{\ell/2}\frac{15^{\ell/2}}{8g \cdot 14^{\ell/2}} \leq \frac{(5n)^{\ell}}{8g} = \frac{|R|}{8g}$.

Therefore, at least $|R|/4g$ demands are routed via heavy answer edges. We now define a strategy for the two provers, as follows. For each query q, we randomly choose one of the at most c heavy answers. The expected number of demands that can still be routed (by using the chosen answer edges) is at least $|R|/4gc^2$. Moreover, since we choose one answer for each query, for each random string $r \in R$, only one demand (r,b) can be routed. Thus, we route at least $|R|/4gc^2$ demands, belonging to different random strings, and all these random strings are satisfied by the chosen answers. If we ensure that $4gc^2 < 2^{\alpha\ell}$, where α is the constant of the Raz verifier, we get a contradiction.

□

Lemma 6.3 *With high probability, there is at least one edge with congestion c.*

Proof: Fix a good pair (y,z), and let q be the heavy query for this pair. We define a bad event $\eta(y,z)$ when no edge in the blob $B(q,y,z)$ has congestion c. We now bound the probability of $\eta(y,z)$. Denote the heavy answers by $a_1,\ldots,a_c$. For each heavy answer a_i, there are $x = \frac{15^{\ell/2}}{14^{\ell/2} \cdot 8g}$ paths in the solution that use edge $e(q,a_i)$ in graph P, and traverse the blob $B(q,y,z)$ in graph T.

We consider the first $x/2$ edges in blob $B(q,y,z)$, denoted by $e_1,\ldots,e_{x/2}$. For each such edge e_j, $1 \leq j \leq x/2$, regardless of the configuration of previous edges, the probability that c of the paths that correspond to heavy answers are mapped to this edge is at least $(x/2I)^c$. Thus, the probability of the event $\eta(y,z)$ happening is bounded by:

$$\left(1-\left(\frac{x}{2I}\right)^c\right)^{x/2} \leq e^{-\frac{x}{2}\left(\frac{x}{2I}\right)^c}$$

Let η be the event that for all good pairs (y,z), the event $\eta(y,z)$ happened. Notice that if η does not happen, then we have at least one edge with congestion c. Since there are $|Y|Z/2g$ good pairs (y,z), the probability of η happening is:

$$e^{-\frac{|Y|Z}{2g}\cdot\frac{x}{2}\left(\frac{x}{2I}\right)^c}$$

The number of possible choices of a set $\mathcal{S}$ containing $|Y||R|/g$ paths is:

$$\binom{|Y||R|7^{\ell}}{|Y||R|/g} \leq (ge \cdot 7^{\ell})^{|Y||R|/g} \leq e^{O(\ell|Y||R|/g)}$$

(since $g \leq 2^{\alpha\ell}/c^2$). We now use the union bound. To ensure that the probability that there is no congested edge is small, it is enough to have:

$$Z = O\left(\ell|R|\left(\frac{2I}{x}\right)^c\right) = O\left(\ell|R|\left(\frac{2\cdot 8^\ell \cdot 8g\cdot 14^{l/2}}{15^{\ell/2}}\right)^c\right) = 2^{O(\ell c)}\cdot 2^{O(\ell\log n)}$$

□ □

6.4 Hardness Factor

Let $c \geq \log n$ be the congestion factor. For this choice of c, note that $N = O(Z^3)\cdot n^{O(\ell)} = 2^{O(\ell c)}$ and the parallel repetition parameter $\ell = \Theta\left(\frac{\log N}{c}\right)$. The hardness gap factor g is maximized by setting $g = 2^{\alpha\ell}/8c^2$. For any $\lambda > 0$, by varying the parameter ℓ between $\Theta\left(\log^{\frac{1}{\lambda}-1} n\right)$ and $\Theta(\log\log n)$, we vary the allowed congestion c to be a desired function of N that satisfies $\Theta(\log^\lambda N) \leq c \leq \Theta(\log N/\log\log N)$. Substituting the value of ℓ and c in the expression $g = 2^{\alpha\ell}/8c^2$, we get the following theorem.

Theorem 6.4 *There exists a constant $\delta > 0$, such that for any constant $0 < \lambda < 1$, directed EDP with congestion c is hard to approximate to within a factor of $N^{\Omega(1/c)}$, unless $NP \subseteq ZPTIME(n^{\text{polylog(n)}})$, for congestion $c : \log^\lambda N \leq c \leq (\delta\log N)/(\log\log N)$. This result holds even on unique paths instances.*

7 Congestion Minimization

We now prove the hardness of directed congestion minimization.

Construction

Our construction is a simple modification of the construction of the previous section. We start with the instance constructed above, and for each random string $r \in R$ and for each label $y \in Y$, we create a new source $s(r, y)$ and a new sink $t(r, y)$. The source-sink pairs in the new graph, say H, are only these newly added source-sink pairs of the form $s(r, y)$-$t(r, y)$. We now connect these new source-sink pairs to the original source-sink pairs as follows. Consider any assignment $b \in A(r)$. Let y_b be the unique label, such that $y_b = y + u(r, b)$. Then we connect $s(r, y)$ to the original source $s(r, b, y_b)$. Additionally, let $t(r, b, y_b)$ be the sink corresponding to source $s(r, b, y_b)$ in the original graph. Then we connect $t(r, b, y_b)$ to $t(r, b)$.

Observe now that for each source-sink pair in H, there are 7^ℓ canonical paths: these are the canonical paths corresponding to each one of the source-sink pairs in the original graph, to which the new source-sink pair is connected. Each such canonical path represents one assignment $b \in A(r)$.

Yes-Instance Analysis

For the YES-INSTANCE, observe that the solution used in the YES-INSTANCE of EDPwC is a feasible solution here: each source-sink pair $s(r, y)$-$t(r, y)$, is routed on the canonical path corresponding to source-sink pair $s(r, b, y_b)$-$t(r, b, y_b)$, where b is the "correct" assignment to the variables corresponding to r. This is a solution that routes all the source-sink pairs and its congestion is 1.

No-Instance Analysis

For analyzing the No-Instance, it suffices to prove that each source-sink pair in the new graph can be routed only along one of the 7^{ℓ} canonical paths. We can then directly apply the gap analysis of the preceding section.

Lemma 7.1 *Let $s(r,y)$-$t(r,y)$ be any source-sink pair in the new graph, and let P be any path connecting $s(r,y)$ to $t(r,y)$. Then there is an assignment $b \in A(r)$, such that after leaving $s(r,y)$ and before arriving to $t(r,y)$, P follows the unique canonical path P' connecting $s(r,b,y_b)$ and $t(r,b,y_b)$.*

Proof: Assume otherwise. Let $t(r,b,y_b)$ be the sink in the original graph that P visits before reaching $t(r,y)$. Let P' be the unique canonical path connecting $s(r,b,y_b)$ to $t(r,b,y_b)$, and let y' be the last label along the path P' before it reaches $t(r,b,y_b)$. Then $y' = y + (Z+1)u(r,b)$. If P does not follow P', then at some point it must use an increment vector u different from $u(r,b)$. Let $u^1,\ldots,u^{Z+1}$ be the sequence of increment vectors used by P', where for some $i : 1 \le i \le Z+1$, $u^i \neq u(r,b)$. Then $y' = y + u^1 + \cdots + u^{Z+1}$. Following the same reasoning as in the proof of Corollary 1 and Claim 3.1, this is impossible. □

We can now use the No-Instance analysis from the previous section to show that if more than C_{YI}/g paths are routed, then with high probability there is an edge with congestion c. Setting $c = \log n$ and $\ell = \Theta(\log\log n)$, we obtain construction size $N = 2^{O(\ell c)} = n^{O(\log\log n)}$, while the hardness gap is $c = \Omega(\log N/\log\log N)$. We thus get the theorem below.

Theorem 7.1 *Congestion minimization in directed graphs is $\Omega(\log N/\log\log N)$-hard to approximate unless $NP \subseteq ZPTIME(n^{\log\log n})$. Moreover, there exists a constant $\delta > 0$ such that even for instances where the optimal solution has congestion 1, there is no poly-time algorithm that finds a solution with congestion less than $\delta \log N/\log\log N$ unless $NP \subseteq ZPTIME(n^{\log\log n})$.*

8 Extensions

In all our hardness results for EDPwC, each source-sink pair has a unique path along which it can be connected. As a result, all our results also hold for the All-or-Nothing flow problem, a relaxation of the EDPwC problem, where instead of choosing a single path connecting e, one is required to route one unit of flow between them.

It is also easy to see that all our results extend to the vertex versions of the respective problems, where the congestion is measured on vertices and not edges of the input graph. In each one of our constructions, we can add a special vertex in the middle of each special edge (i.e., each special edge is replaced by a path of two edges). It is easy to see that the maximum congestion is always obtained on the special vertices, and that the solution values for the edge and the vertex versions are the same.

Finally, we notice that all our constructions are directed acyclic graphs, and thus all our hardness of approximation results hold on directed acyclic graphs as well.

Acknowledgement

We would like to thank Chandra Chekuri for many helpful comments, and Piotr Indyk for pointing us to the field of additive combinatorics.

References

[1] M. Andrews. Hardness of Buy-at-Bulk Network Design. *Proc. of FOCS*, pp. 115-124, 2004.

[2] M. Andrews and L. Zhang. Hardness of the undirected edge-disjoint paths problem. *Proc. of STOC*, 2005.

[3] M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. *Proc. of STOC*, 2005.

[4] M. Andrews and L. Zhang Logarithmic hardness of the directed congestion minimization problem. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, 2006, pages 517–526.

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

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[7] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *JACM*, 45(1):70–122, 1998.

[8] Y. Azar and O. Regev. Strongly Polynomial Algorithms for the Unsplittable Flow Problem. IPCO 2001: 15-29.

[9] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *Mathematics of Operations Research*, Vol. 25, pp. 255–280, 2000.

[10] C. Chekuri, S. Khanna, and F. B. Shepherd. The All-or-Nothing Multicommodity Flow Problem. *Proc. of STOC*, 2004.

[11] C. Chekuri, S. Khanna, and F. B. Shepherd. Multicommodity Flow, Well-linked Terminals, and Routing Problems. *Proc. of STOC*, 2005.

[12] J. Chuzhoy and S. Naor. The Hardness of Metric Labeling. In *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004, pp. 108-114.

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

[14] V. Guruswami and K. Talwar. Personal communication. May 2006.

[15] J. Håstad. Clique is hard to approximate to within $n^{1-\epsilon}$. In *Proceedings of the 37th Symposium on Foundations of Computer Science*, pages 627–636, 1996.

[16] S. Khot. Improved Inapproximability Results for Max Clique, Chromatic Number, and Approximate Graph Coloring. *Proc. of FOCS*, pp. 600–609, 2001.

[17] P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[18] R. Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27:3, pp. 763–803, 1998.

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

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/