# Hardness of Routing with Congestion in Directed Graphs

Julia Chuzhoy[*]
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
cjulia@math.ias.edu

Venkatesan Guruswami[†]
Dept. of Comp. Sci. & Engg.
University of Washington
Seattle, WA 98195
venkat@cs.washington.edu

Sanjeev Khanna[‡]
Dept. of Comp. and Inf. Sci.
University of Pennsylvania
Philadelphia, PA 19104
sanjeev@cis.upenn.edu

Kunal Talwar
Microsoft Research
Silicon Valley Campus
Mountain View, CA 94043
kunal@microsoft.com

## ABSTRACT

Given as input a directed graph on $N$ vertices and a set of source-destination pairs, we study the problem of routing the maximum possible number of source-destination pairs on paths, such that at most $c(N)$ paths go through any edge. We show that the problem is hard to approximate within an $N^{\Omega(1/c(N))}$ factor even when we compare to the optimal solution that routes pairs on edge-disjoint paths, assuming NP doesn't have $N^{O(\log\log N)}$-time randomized algorithms. Here the congestion $c(N)$ can be any function in the range $1 \leqslant c(N) \leqslant \alpha \log N/\log\log N$ for some absolute constant $\alpha > 0$. The hardness result is in the right ballpark since a factor $N^{O(1/c(N))}$ approximation algorithm is known for this problem, via rounding a natural multicommodity-flow relaxation. We also give a simple integrality gap construction that shows that the multicommodity-flow relaxation has an integrality gap of $N^{\Omega(1/c)}$ for $c$ ranging from 1 to $\Theta(\frac{\log n}{\log\log n})$.

A solution to the routing problem involves selecting which pairs to be routed and what paths to assign to each routed pair. Two natural restrictions can be placed on input instances to eliminate one of these aspects of the problem complexity. The first restriction is to consider instances with *perfect completeness*; an optimal solution is able to route all pairs with congestion 1 in such instances. The second restriction to consider is the *unique paths* property where each source-destination pair has a unique path connecting it in the instance. An important aspect of our result is that it holds on instances with *any one* of these two restrictions. Our hardness construction with the perfect completeness restriction allows us to conclude that the directed congestion minimization problem, where the goal is to route all pairs with minimum congestion, is hard to approximate to within a factor of $\Omega(\log N/\log\log N)$. On the other hand, the hardness construction with unique paths property allows us to conclude an $N^{\Omega(1/c)}$ inapproximability bound also for the all-or-nothing flow problem. 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.

**Categories and Subject Descriptors:** F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems

**General Terms:** Algorithms, Theory.

**Keywords:** All-or-Nothing Flow, Congestion Minimization, Edge-disjoint Paths, Hardness of approximation, Integrality Gap, Multicommodity Flow.

[*]Supported by a grant of the state of New Jersey to the Institute for Advanced Study.

[†]Supported by NSF CCF-0343672 and a Sloan Research Fellowship.

[‡]Supported in part by an NSF Career Award CCR-0093117 and by NSF Award CCF-0635084.



## 1. INTRODUCTION

We study the approximability of routing problems with bounded congestion in directed graphs. The input to all problems in our study is a directed graph on $N$ vertices and a collection of source-destination pairs. When the goal is to route a maximum number of pairs with the restriction that at most $c(N)$ paths go through any edge, we refer to the problem as DirEDPwC (directed edge-disjoint paths with congestion). The parameter $c(N)$ is refered to as the *congestion*. When $c(N) = 1$, we get the classical *maximum edge-disjoint paths* problem. If we relax the notion of routing in DirEDPwC to be a fractional routing of a unit of flow from a source to its destination, we get the DirANFwC (directed all-or-nothing flow with congestion) problem. Finally, if the objective is to route all source-destination pairs with minimum congestion, we get the *congestion minimization* problem.

Our main result is that DirEDPwC is hard to approximate within an $N^{\Omega(1/c(N))}$ factor for all integer valued functions $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(N) \leqslant \alpha \log N/\log\log N$, even when compared to an optimal solution that routes with congestion 1. Here $\alpha > 0$ is an absolute constant, and the hardness holds under the assumption that NP $\not\subseteq$ BPTIME($N^{O(\log\log N)}$). An important aspect of our inapproximability result is that it holds even on two restricted classes of input instances: on instances with *perfect completeness* and on instances with the *unique paths* property. Specifically, the perfect completeness property means that our result holds even on input instances where all the source-destination pairs can be routed on *edge-disjoint paths* (i.e., with no congestion). In other words, it is hard to find a good solution with congestion $c(N)$ even if a (perfect) solution with congestion 1 exists. As a corollary, we get an $\Omega(\log N/\log\log N)$ hardness for the *directed congestion minimization* problem, where one must route all the pairs and the quality of the solution is measured by the maximum congestion incurred on any edge. This is a slight improvement to the recent factor $\Omega(\log^{1-\varepsilon} N)$ hardness shown by Andrews and Zhang [4]. But the nice aspect is that this inapproximability factor is tight up to constant factors since a classic randomized rounding based algorithm of Raghavan and Thompson [21] achieves an approximation ratio of $O(\log N/\log\log N)$ for the congestion minimization problem. A second restriction, under which our hardness result holds, is the unique paths property which means that each source-destination pair is connected by a *unique path* in the input instance. As a corollary, we get an $N^{\Omega(1/c(N))}$-hardness for DirANFwC. This is in a sharp contrast to the undirected case where an $O(\log^2 N)$ approximation ratio is known even with congestion 1 [6, 7].

We note that all problems in our study are trivially solvable in polynomial-time on instances that simultaneously possess the perfect completeness and the unique paths property.

## 1.1 Prior Work

When the congestion $c = 1$, DirEDPwC is identical to the classic edge-disjoint paths problem (on directed graphs). This was shown to be hard to approximate within a $N^{1/2-\varepsilon}$ factor for any $\varepsilon > 0$ [12], and a natural greedy algorithm is known to give an $O(\min\{N^{2/3}, \sqrt{M}\})$-approximation (where $M$ is the number of edges) [19, 5, 23].

For the DirEDPwC problem with congestion $c > 1$, factor $O(cN^{1/c})$ approximation algorithms based on randomized rounding of the standard multicommodity flow relaxation are known [22, 20]. Hardness results for DirEDPwC with congestion $c > 1$ have been elusive until recently. The reduction in [12] does not give any hardness result once $c$ exceeds 1. Spurred by the breakthrough results of Andrews and Zhang on undirected edge-disjoint paths and congestion minimization [3, 2], a factor $(\log N)^{\Omega(1/c)}$ hardness for the above problem on *undirected graphs* has been shown [1, 13]. This hardness also holds for directed graphs.

For directed congestion minimization problem, Chuzhoy and Naor [9] were the first to obtain a non-trivial inapproximability result – they established a factor $\Omega(\log\log N)$ hardness result. Prior to [9], even a factor 2 hardness was not known (a factor $(2-\varepsilon)$ hardness is trivial since it is NP-hard to tell if all source-destination pairs can be routed in an edge-disjoint manner).

## 1.2 Our Results and Reduction Techniques

The results reported here are a merged version of results from the two technical reports [8, 14]. The reader can refer to these reports for any details or proofs omitted in this version. We briefly describe the results that appear in these reports. In [8], two hardness of approximation results are presented. The first result shows $N^{1/(3c+O(1))}$-hardness for DirEDPwC for values of $c$ up to $\log^{\lambda} N$ for some suitably small $\lambda$. This construction has the unique paths property but not perfect completeness. We note that the unique paths property automatically gives a similar hardness of approximation result for DirANFwC. The second result is an $N^{\Omega(1/c)}$-hardness for values of $c$ ranging from $\log^{\lambda} N$ to $\Omega(\log N/\log\log N)$ for any $0 < \lambda < 1$. This result also has the unique paths property, and moreover, [8] shows a simple transformation that achieves perfect completeness by trading off the unique paths property. Consequently, this establishes an $\Omega(\log N/\log\log N)$-hardness for congestion minimization. In [14], a reduction from general constraint satisfaction problem to DirEDPwC is presented that gives an $N^{\Omega(1/c)}$-hardness for values of $c$ up to $\Omega(\log N/\log\log N)$. This reduction has the perfect completeness property, and thus gives as a corollary $\Omega(\log N/\log\log N)$ hardness for congestion minimization. For large values of $c$ this construction can be modified to give the unique paths property.

A common theme in the recent hardness proofs for routing problems is the idea of "canonical" and "non-canonical" paths. Roughly speaking, each source-destination 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, and when the solutions are restricted to choose canonical paths only, the problem is shown to be hard to approximate. The main obstacle to proving hardness of approximation for this type of routing problems is therefore the existence of non-canonical paths. As a way to overcome this problem in the context of directed graphs, in a beautiful work, 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-destination 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 $\Omega\left(\log^{1-\varepsilon} N\right)$-hardness for directed congestion minimization, for any *constant* $\varepsilon > 0$. It is also possible to adapt their ideas in a straightforward manner to show that DirEDPwC is $2^{\Omega(\log^{\frac{1}{2}-\varepsilon} N)}$-hard to approximate for constant congestion.

A key idea underlying all our results is a more efficient labeling scheme, which is described in Definition 2.6. This labeling scheme combined with the framework of [4] would suffice for our hardness result for DirEDPwC when the congestion is at least $\log^{\varepsilon} n$ for some $\varepsilon > 0$. For smaller values of congestion, one needs to use a different approach. For example, for constant congestion $c$, we are seeking a $N^{\Omega(1)}$ hardness, while the soundness of the Raz 2-prover system, which is the starting point for the reduction to congestion minimization in [4] is not polynomially small.

As one approach to overcome this problem, we present a simple reduction from the independent set problem that establishes a $N^{\frac{1}{3c+O(1)}}$ hardness for DirEDPwC with congestion $c$ for $1 \leqslant c \leqslant \log^{\lambda} N$ for some absolute constant $\lambda > 0$. It is easy to show that graphs where the maximum independent set size is small, necessarily contain a large clique. We establish an interesting generalization of this property, namely, such graphs in fact contain many large cliques. Specifically, we show that if an $n$-vertex graph does not have an independent set of size $n^{O(1/c^6)}$, then it contains $n^{\Omega(c)}$ cliques of size $c$. Our reduction translates cliques of size $c$ into edges with congestion $c$ in the DirEDPwC instance. This reduction creates instances with the *unique paths* property, and therefore also gives hardness of DirANF and DirANFwC.

Our result with perfect completeness is based on a reduction from general constraint satisfaction problems over large domains. Conceptually, our reduction is presented in a rather general framework and can start from an arbitrary constraint satisfaction problem (CSP). We analyze the efficacy of the reduction in this extreme generality, and quantify its performance in terms of few crucial parameters of the original CSP. We then plug in appropriate CSPs as starting point to deduce our hardness results. The benefit of this unified approach is that it clearly highlights what one needs as a starting point from the CSP in order to obtain strong hardness results for DirEDPwC. For example, when the congestion is a constant, we are seeking an $N^{\Omega(1)}$ inapproximability factor. As we mentioned above, in order to obtain this one needs to start from a hard CSP with soundness that is inverse polynomial in the size of the instance. The 2-variable CSP underlying Label Cover (or Raz's parallel repetition theorem), which formed the starting point of the reduction in [4], does not achieve such small soundness. A crucial (but quite natural) technique we use is to boost the soundness of a hard CSP using a derandomized expander-walk based serial repetition. This results in the low soundness CSP which we then plug into our general reduction to DirEDPwC. Doing this to the standard 2-variable "Label Cover" CSP gets us a *unified* $N^{\Omega(1/c(N))}$ inapproximability factor for DirEDPwC for the *entire range* $1 \leqslant c(N) \leqslant O(\log N/\log\log N)$. The instances produced by this reduction do not have the unique paths property, and they *cannot*, since the problem with perfect completeness and unique paths property is of course trivial. But with a small modification to the construction, we can also get unique paths instances and show a $N^{\Omega(1/c(N))}$ for congestions larger than $\log^{\Omega(1)} N$. Together with the reduction from independent set, we get the hardness on unique paths instances, and hence also for ANFwC, for the entire range $[1, \Theta(\log N/\log\log N)]$ of congestions.

The drawback of the above reduction from general CSPs is that the constant in the exponent $\Omega(1/c(N))$ is rather small as it is inherited from the constant in Raz's parallel repetition theorem. In order to improve the constant, we start from a different 3-variable CSP. For us, the crucial feature of this CSP is that the maximum number of satisfying assignments to any of its constraints is comparable to the inverse of the soundness of the CSP. This gives an improved $N^{1/((9+\varepsilon)(c(N)+6))}$ hardness factor for any desired constant $\varepsilon > 0$. However, the size of this 3-variable CSP, for a similar size of the domain of the variables, is somewhat larger than the Raz system. This limits the range of applicability of the improved bound to $1 \leqslant c(N) \leqslant \log^{\beta} N$ for some $\beta > 0$ that depends on $\varepsilon$. The hardness result still holds with perfect completeness.

If we start from an even "better" CSP, our reduction will yield a factor $N^{\frac{1}{(3+\varepsilon)(c(N)+O(1))}}$ hardness result. The CSPs shown to hard to approximate by Håstad and Khot [17] have the correct parameters to function as our needed starting point, except that they are not $k$-partite (where each constraint depends on $k$ variables) which is a structural feature we find useful in our analysis. We feel that hardness results for CSPs similar to those proved in [17] should continue to hold even with the restriction of $k$-partiteness.

## 1.3 Organization

We begin with some background and definitions in Section 2; in particular, we define the problems and their variants that are of interest to us, review the natural multicommodity flow LP relaxation for DirEDPwC, and also define the labeling scheme that will be crucial in our reductions. Using this labeling scheme, we prove a strong integrality gap result for the natural flow relaxation of DirEDPwC in Section 3.

In Section 4, we give a reduction from the maximum independent set problem to DirEDPwC. This reduction yields a hardness of $N^{\frac{1}{3c+23}}$ for low congestion values (less than $\log^{\lambda} N$ for some constant $\lambda > 0$). This reduction additionally gives *unique path* instances: i.e., for each source-destination pair in the instance, there is a unique path from the source to the corresponding destination in the graph. Consequently, this immediately yields similar hardness results for DirANF and DirANFwC.

In Section 5, we give a different reduction from a general CSP to DirEDPwC. For an appropriately chosen CSP, this proves a hardness of $N^{\Omega(\frac{1}{c})}$ for large congestion values (all the way up to $\Theta(\log N/\log\log N)$). In addition, this gives instances with perfect completeness, thus also proving an $\Omega(\log N/\log\log N)$ hardness for the congestion minimization problem.

# 2. BACKGROUND AND DEFINITIONS

## 2.1 Problems and their variants

We start with a definition of the problems that we study.

Definition 2.1 (Congestion Minimization). *Given a graph $G(V, E)$ and a collection of source-destination pairs $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$, find a routing that connects each source $s_i$ to its destination $t_i$ by a path, with at most $c$ paths going through any edge, such that the congestion $c$ is minimized.*

Definition 2.2 (DirEDP). *Given a directed graph $G(V, E)$ and a collection of source-destination 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 2.3 (DirEDPwC). *Given a directed graph $G(V, E)$, a collection $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ of source-destination pairs, and an integer $c \geqslant 1$, route as many pairs as possible such that at most $c$ paths go through any edge. The performance of algorithms for* DirEDPwC *is compared to optimal solutions with congestion* 1.

**All-or-Nothing Flow:** DirANF (All-or-Nothing Flow) and DirANFwC (All-or-Nothing Flow with congestion) are defined analogously to DirEDP and DirEDPwC by relaxing the notion of routing: a source-destination pair is routed if the solution sends a unit of flow fractionally from the source to its destination.

We note that all the above 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-destination pairs are distinct; instances created by our reductions will satisfy this property.

**Unique Paths Instance:** We will say an instance of DirEDPwC is a *unique paths* instance if every source-destination pair in the instance has a unique path that connects the source to the destination. Note that a hardness result for DirEDP or DirEDPwC based on a unique paths instance immediately implies the same hardness result for DirANF and DirANFwC respectively.

**Perfect Completeness:** We will say that a factor $f$ inapproximability result for directed EDP with congestion $c$ holds with perfect completeness, if given a set of $(s_i, t_i)$ pairs **all** of which can be routed on **edge-disjoint** paths, i.e., with congestion 1, it is NP-hard to route more than a fraction $1/f$ of the pairs, even if congestion $c$ is allowed. Note that a hardness result for EDP with congestion $c$ that holds with perfect completeness immediately implies a factor $c$ inapproximability result for congestion minimization.

**Multicommodity Flow Relaxation:** Given an instance of DirEDPwC 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 $s_i$-$t_i$. Then the multicommodity flow relaxation for DirEDP is as follows:

$$\max \sum_{i=1}^{k} x_i \quad \text{s.t}$$

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

We use the same multicommodity flow relaxation for DirEDPwC and we are interested in comparing fractional solutions with no congestion to integral solutions with congestion less than $c$. Even though as stated above, the formulation is of exponential-size, there is a standard equivalent flow-based encoding of this linear program that is polynomial-size.

## 2.2 Convexly Independent vectors

Our reductions crucially use a certain labeling scheme to group vertices that prevents paths connecting certain pairs of vertices from taking untoward detours in the graph. The labeling scheme uses a collection of vectors with a restricted linear independence property, defined below.

Definition 2.4. *A set of vectors $u_1, \ldots, u_n$ over reals is* convexly independent *if for any $i$ and for any set of nonnegative coefficients $\alpha_j \geqslant 0$, if $\sum_j \alpha_j = 1$ and $u_i = \sum_{j=1}^n \alpha_j u_j$, then $\alpha_i = 1$ and $\alpha_j = 0$ for $j \neq i$.*

Our goal is to construct a large set of convexly independent vectors in a low dimensional space. Note that we can take $u_i = e_i$ in $\mathbb{R}^n$, and being linearly independent, this set of vectors is also convexly independent. Andrews and Zhang show how to do better: let $u_i$ be a 0-1 vector with $\frac{k}{3}$ 1's in $\mathbb{R}^k$, where $k$ is chosen so that $\binom{k}{\frac{k}{3}} > n$. It is easy to verify that the resulting $u_i$'s are convexly independent and that $k$ is only $O(\log n)$.

We now give a simple construction in two dimensions. Given our goal of "convex" independence, it is natural to turn to one of the simplest convex functions: the squaring function. We omit the easy proof.

Lemma 2.5. *For every integer $n \geqslant 1$, the set of vectors $u_1, u_2, \ldots, u_n \in \mathbb{R}^2$ given by $u_i = (i, i^2)$ is convexly independent.*

Definition 2.6 (Labeling Scheme). *The labeling scheme is parameterized by two integers $A, Z$, and consists of two ingredients:*

1. *A set $U$ of $A$* increment vectors *$u_1, u_2, \ldots, u_A$, each in $\mathbb{Z}^2$, defined by $u_i = (i, i^2)$*
2. *A label set $\mathcal{L} = \mathcal{L}(A, Z)$, together with an addition operator $'+'$ (under which the set is closed), defined as follows: $\mathcal{L} = [2AZ] \times [2A^2 Z]$, where the notation $[M]$ stands for $\{0, 1, \ldots, M-1\}$. The addition operation on $\mathcal{L}$ is defined to be coordinate wise addition w.r.t. the appropriate modulus. Formally, for $(a, b) \in \mathcal{L}$ and $(a', b') \in \mathcal{L}$, their sum $(a, b) + (a', b') \in \mathcal{L}$ is defined to be $((a + a') \mod (2AZ), (b + b') \mod (2A^2 Z))$.*

We now record the crucial property of the above labeling scheme, which follows immediately from Lemma 2.5.

Lemma 2.7. *Let $l, l' \in \mathcal{L}$ be a pair of labels such that there is some $1 \leqslant j \leqslant A$ for which $l + Zu_j = l'$. Let $u_{i_1}, \ldots, u_{i_Z}$ be any collection of $Z$ increment vectors from $U$. Then $l + u_{i_1} + \cdots + u_{i_Z} = l'$ iff $u_{i_1} = u_{i_2} = \cdots = u_{i_Z} = u_j$.*

# 3. INTEGRALITY GAP

We present an elementary construction that shows that the integrality gap of the multicommodity flow relaxation is $\Omega\left(\frac{N^{\frac{1}{3c'+13}}}{c'}\right)$ for directed EDP with congestion $c'$, where $c'$ can be any integer between 1 and $\delta \log N / \log \log N$, for some fixed constant $\delta$. We will use the labeling scheme described above, consisting of a set $U$ of increment vectors and a set $\mathcal{L} = \mathcal{L}(A, Z)$ of labels. We will set the parameter $A = n$; the value of $Z$ will be specified later. Let $\{u_i\}_{i=1}^n$ be the increment vectors in $U$. Let $c = c' + 1$. We also denote $L = |\mathcal{L}|$.

Our instance is a layered graph with $Z$ layers and $nL$ source-destination pairs. For each $y \in \mathcal{L}$, for each $i : 1 \leqslant i \leqslant n$, there is a source-destination pair $s(y, i)$-$t(y, i)$, and we will later define the corresponding canonical path $P(y, i)$. For each layer $z : 1 \leqslant z \leqslant Z$, for each label $y \in \mathcal{L}$, we have a set $E(y, z)$ of $n/c$ *special edges*, whose endpoints are disjoint.

A canonical path $P(y, i)$ starts at source $s(y, i)$ and then traverses, for each $z : 1 \leqslant z \leqslant Z$ one special edge at layer $z$. After traversing a special edge at layer $Z$, it finishes at destination $t(y, i)$. 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 $z: 1 \leqslant z \leqslant Z$. Fix some $y \in \mathcal{L}$, $i: 1 \leqslant i \leqslant n$. 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 $y \in \mathcal{L}$. 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 $i: 1 \leqslant i \leqslant m$, there is exactly one such path $P(y',i)$, and thus $|\mathcal{P}_z(y)| = n$. 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. Notice that the size of the construction is $N \leqslant O(nZL) = O(nZ \cdot 4n^3Z^2) = O(n^4Z^3)$.

**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 $nL/c$ units of flow.

**Integral Solution:** We now proceed to analyze integral solution. Consider any integral solution with congestion strictly less than $c$. We will show that it routes at most $nL/cg$ pairs where $g$ will be the integrality gap, and we specify its value later.

LEMMA 3.1. *For each source-destination pair in our construction, there is only one path connecting the source to the destination (the canonical path).*

PROOF. Consider any source-destination 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 $k: 1 \leqslant k \leqslant Z$, $y_k = y_{k-1} + u_{j_k}$. 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 $u_{j_p}$, $1 \leqslant p \leqslant Z$ differs from $u_i$. But then, by Lemma 2.7, it is impossible that $P'$ reaches $t(y,i)$. $\square$

LEMMA 3.2. *Suppose $Z \geqslant (8c)^{c+3}g^{c+2}$ and $g \leqslant n/8c^2$. Then if more than $nL/cg$ pairs are routed, then at least one edge has congestion $c$, with high probability.*

PROOF. Let $\mathcal{S}$ be any collection of at least $nL/cg$ canonical paths. We say a label $y$ at a layer $z$ is *good* if there are at least $n/2cg$ paths $P(y',i) \in \mathcal{S}$, which have label $y$ at layer $z$. At least $L/2cg$ labels must be good at each layer $z$. Otherwise, total number of paths contained in $\mathcal{S}$ is *strictly less than* $\left(\frac{L}{2cg}\right)n + L\left(\frac{n}{2cg}\right) = \left(\frac{nL}{cg}\right)$, a contradiction! Thus there must be at least $(L)/2cg$ good labels at each layer.

Consider a good label $y$ at a layer $z$. We say that the *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 $n/4c^2g$ edges $e_1, \ldots, e_{n/4c^2g}$ in $E(y,z)$, and assume that each edge chooses a $c$-tuple of paths in this order. For each $j: 1 \leqslant j \leqslant n/4c^2g$, 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:

$$\frac{\binom{\frac{n}{4cg}}{c}}{\binom{n}{c}} \geqslant \left(\frac{\frac{n}{4cg} - c + 1}{n}\right)^c \geqslant \left(\frac{1}{8cg}\right)^c$$

Thus the probability that $B(y,z)$ occurs is at most

$$\left(1 - \frac{1}{(8cg)^c}\right)^{n/4c^2g} \leqslant e^{-\frac{n}{(4c^2g)(8cg)^c}} \leqslant e^{-\frac{n}{(8c)^{c+2}g^{c+1}}}.$$

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 $z$. Using the assumption $Z \geqslant (8c)^{c+3}g^{c+2}$, we get

$$\mathbf{Pr}[B \text{ occurs}] \leqslant e^{-\frac{nLZ}{2cg(8c)^{c+2}g^{c+1}}} \leqslant e^{-nL}.$$

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

Recall that the construction size is $N \leqslant O(n^4Z^3)$. Substituting $Z = (8c)^{c+3}g^{c+2}$ and $g = n/8c^2$, we get construction size $N \leqslant O(n^4(8c)^{3c+9}g^{3c+6}) \leqslant O((8c^2)^4(8c)^{3c+9}g^{3c+10}) \leqslant O((8c)^{3c+20}g^{3c+10})$. Thus $g = \Omega(\frac{N^{\frac{1}{3c+10}}}{c})$, and we obtain the following theorem.

THEOREM 3.3. *There exists a constant $\delta > 0$, such that the integrality gap of the multicommodity flow relaxation for* DirEDPwC *is $\Omega(N^{\frac{1}{3c'+13}}/c')$ for any congestion value $c'$, $1 \leqslant c' \leqslant (\delta \log N)/(\log\log N)$, where $N$ is the number of vertices in the graph.*

## 4. HARDNESS OF DirEDPwC WITH UNIQUE PATHS PROPERTY

In this section we show that DirEDPwC is hard on instances with unique paths property. We present here only the hardness result for congestion values up to $\log^\lambda N$, for some absolute positive constant $\lambda$.

We start by establishing hardness for any constant congestion $c'$, and later show how it can be extended to higher values of $c'$. The hardness applies to unique paths instances of DirEDPwC and therefore immediately implies a similar hardness for All-or-Nothing flow with congestion as well.

**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 4.1. *[15] For any $\varepsilon > 0$, no polynomial-time algorithm can distinguish between $n$-vertex graphs that have an independent set of size at least $n^{1-\varepsilon}$* (the YES-INSTANCE) *and graphs that have no independent sets of size greater than $n^\varepsilon$* (the NO-INSTANCE), *unless NP is contained in ZPP.*

Let $c = c' + 1$. 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 $\Omega(1/n^\varepsilon)$-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 $\varepsilon < \frac{1}{2c^6}$.

We will use the labeling scheme of Definition 2.6 with parameters $A, Z$. We will set $A = n$, where $n$ is the number of vertices in the independent set instance, and the parameter $Z$ will be specified later. We will denote by $U$ the set of increment vectors, and by $\mathcal{L} = \mathcal{L}(n,Z)$ the set of labels. As in the integrality gap construction above, we build a layered graph with $Z$ layers. For each layer $z : 1 \leqslant z \leqslant Z$, and for each label $y \in \mathcal{L}$, we have $n/c$ blobs $B(y, z, i)$ of special edges, where $1 \leqslant i \leqslant n/c$. Each blob will either contain one special edge, or $c$ special edges. This is determined as follows. For each label $y \in \mathcal{L}$, for each layer $z \in Z$, we choose a random partition $\mathcal{P}_1(y, z), \dots, \mathcal{P}_{n/c}(y, z)$ of the vertices in $V(G)$ into $c$-tuples. For each $i : 1 \leqslant i \leqslant n/c$, 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 $y \in \mathcal{L}$, there is a source-destination 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 $z : 1 \leqslant z \leqslant Z$, 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 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 $z : 1 \leqslant z \leqslant Z$, for each $1 \leqslant v \leqslant m$ and for each $y \in \mathcal{L}$, the label of $P(y, v)$ at layer $z$ to be $y + (zu_v)$. At each layer $z : 1 \leqslant z \leqslant Z$, 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. Let $N$ denote the size of the instance $I$ constructed above. Then $N = O(ZLn) = O(Z^3 n^4)$.

**Yes-Instance:** If $G$ is a YES-INSTANCE, then it has an independent set $S$ of size at least $n^{1-\varepsilon}$. For each $v \in S$ and $y \in \mathcal{L}$, we route the source-destination pair $s(y, v)$-$t(y, v)$ on its canonical path $P(y, v)$. This gives a set of $L|S| \geqslant Ln^{1-\varepsilon}$ edge disjoint paths.

**No-Instance:** We show that any subset $\mathcal{M}$ of $4L\sqrt{n}$ source-destination paths causes congestion $c = b + 1$, with high probability. The next lemma follows immediately from the construction and Lemma 2.7.

LEMMA 4.2. *For each source-destination pair in the above instance, the associated canonical path is the unique path connecting them.*

In what follows, let $s = n^{\varepsilon}$. Let $T(\alpha, c)$ denote the minimum number of $c$-cliques in any graph on $\alpha$ vertices, which does not contain an independent set of size $s'$. We need the following technical fact.

LEMMA 4.3. *Let $c \geqslant 2$ be a positive integer. Then for any $\alpha > (4s')^c$, $T(\alpha, c) \geqslant \frac{\alpha^c}{(2c)^c (4s')^{c^3}}$.*

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 $\alpha$ 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 $d = \alpha/2s' - 1 \geqslant \alpha/4s'$: 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 $d \geqslant \alpha/4s' > (4s')^{c-1}$. 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 $T(\alpha, c) \geqslant \frac{\alpha}{2c} T(\alpha/4s', c-1)$. Iterating, we get

$$\begin{aligned} T(\alpha, c) &\geqslant \frac{\alpha}{2c} \cdot \frac{\alpha/4s'}{2(c-1)} \cdot \frac{\alpha/(4s')^2}{2(c-2)} \cdots \frac{\alpha/(4s')^{c-3}}{2(3)} T(\alpha/(4s')^{c-2}, 2) \\ &\geqslant \frac{\alpha^{c-2}}{(2c)^{c-2}(4s')^{c^2/2}} T(\alpha/(4s')^{c-2}, 2) \\ &\geqslant \frac{\alpha^c}{(2c)^c (4s')^{c^3}}. \end{aligned}$$

□

Using the above lemma with the facts that $\varepsilon < 1/c^6$ and $c < \log^{\lambda} N$ for some small $\lambda < 1/4$, we can conclude:

LEMMA 4.4. *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 $c$.*

Assume now that there is a solution that routes a set $\mathcal{M}$ of $4L\sqrt{n}$ 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 $2\sqrt{n}$. It is easy to see that for each layer $z : 1 \leqslant z \leqslant Z$, the fraction of labels $y$ which are good for layer $z$ is at least $2/\sqrt{n}$: Assume otherwise. The number of paths of $\mathcal{M}$ that go through good labels at layer $z$ is at most $2Ln/\sqrt{n} = 2L\sqrt{n}$. The number of paths of $\mathcal{M}$ that go through non-good labels is less than $L \cdot 2\sqrt{n}$. Thus, in total, $\mathcal{M}$ contains less than $4L\sqrt{n}$ 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 $c$. The lemma below bounds the probability of this bad event.

LEMMA 4.5. *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 $y$, $|S| \geqslant 2\sqrt{n}$. 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 $i$th choice is made, if $S_i$ denotes all the vertices chosen at steps $1, \dots, i-1$, 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:

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

□

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

If we set $Z = c^{c+1} n^{\frac{c}{2}+1+\frac{1}{c^2}}$, then this probability is bounded by $e^{Ln}$. The total number of possible solutions is at most $2^{Ln} = o\left(e^{Ln}\right)$. 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 $n^{\frac{1}{2}-\varepsilon}$, while the construction size is $N = O(Z^3 n^4) = O(c^{3c+3} n^{\frac{3c}{2}+8})$. The gap is thus $\Omega\left(N^{\frac{1}{3c+19}}/c\right)$, since $\varepsilon < 1/c^6$. Note that for $c \leqslant \log^{1/4} N$, $\left(N^{\frac{1}{3c+19}}/c\right)$ is $\Omega\left(N^{\frac{1}{3c+20}}\right)$. We thus get the following theorem.

Theorem 4.6. *For any fixed positive constant integer $c' \geqslant 1$, directed EDP with congestion $c'$ is hard to approximate within a factor of $\Omega\left(N^{\frac{1}{3c'+23}}\right)$ unless NP $\subseteq$ ZPP.*

If we slightly relax the complexity assumptions of Theorem 4.1, we can use the theorem below due to Khot [18]:

Theorem 4.7. *[18] 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($n^{\mathrm{polylog}(n)}$).*

Theorem 4.8. *There exists a constant $0 < \lambda < 1/4$ such that for any integer $1 \leqslant c \leqslant (\log N)^{\lambda}$, directed EDP with congestion $c$ on an $N$-vertex graph is hard to approximate within a factor of $\Omega\left(N^{\frac{1}{3c+23}}\right)$ unless NP $\subseteq$ ZPTIME($n^{\mathrm{polylog}(n)}$). Moreover, the hardness holds on Unique Paths instances of* DirEDPwC*, and thus also for directed* ANFwC.

Since *undirected* ANF admits an $O(\log^2 N)$ approximation ratio [6, 7], the theorem above gives a strong separation between the tractability of undirected and directed versions of the all-or-nothing flow problem.

# 5. HARDNESS RESULT FOR DirEDPwC WITH PERFECT COMPLETENESS

In this section, we describe a reduction from general constraint satisfaction problems to DirEDPwC, and instantiate it with particular CSPs. This will give us inapproximability results for DirEDPwC with perfect completeness, and as a corollary a tight (up to constant factors) hardness of approximation result for congestion minimization.

## 5.1 Constraint Satisfaction Problems

Our hardness result for DirEDPwC is obtained via a reduction from a general constraint satisfaction problem (CSP). Informally, a CSP over a domain $\mathcal{D}$ consists of a collection of constraints on a universe of variables, where each constraint specifies which subset of values (from the domain $\mathcal{D}$) to its variables "satisfy" it. The goal is to assign values from $\mathcal{D}$ to all the variables in a manner that maximizes the number of satisfied constraints.

We now give a formal definition that captures the important parameters of a CSP.

Definition 5.1 (General CSP). *An instance of a (promise) constraint satisfaction problem (CSP) with parameters $(M, V, J, p, k, \mathsf{sat}_y, \mathsf{sat}_n)$ where $M, V, J, p, k : \mathbb{N} \to \mathbb{N}$ are integer valued functions, and $\mathsf{sat}_y, \mathsf{sat}_n : \mathbb{N} \to [0,1]$, consists of the following:*

- *A set $S$ of at most $V$ variables that take on values from the domain $\{1, 2, \ldots, p\}$, and a partition of $S$ into $k$ disjoint parts as $S = S_1 \cup S_2 \cup \cdots S_k$.*
- *A set $\mathcal{C}$ of at most $M$ constraints, where each constraint is defined on a subset of $k$ variables containing one variable from each of $S_1, S_2, \ldots, S_k$, and further, at most $J$ of the $p^k$ possible assignments to those variables satisfy that constraint.*

*The goal is, given such an instance, to distinguish between the following two cases:*

- *[Yes instances:] There exists an assignment to the variables that satisfies at least a fraction $\mathsf{sat}_y$ of the constraints. (Note that when $\mathsf{sat}_y = 1$, such an assignment satisfies all the constraints, and this is referred to as* perfect completeness.*)*
- *[No instances:] Every assignment to the variables satisfies less than a fraction $\mathsf{sat}_n$ of the constraints.*

*The parameters $\mathsf{sat}_y$ and $\mathsf{sat}_n$ are called the* completeness *and* soundness *of the CSP. We will sometimes refer to the parameter $p$ as the alphabet size of the CSP, and also call the CSP with above parameters a $p$-ary $k$-partite CSP instance when the rest of the parameters are implicit.*

## 5.2 The Reduction

We will show how to transform an instance of a CSP with parameters $(M, V, J, k, p, \mathsf{sat}_y, \mathsf{sat}_n)$ to an instance of DirEDPwC. Thus our starting point is an instance $\phi$ of a $p$-ary $k$-partite CSP with a set $\{C_1, C_2, \ldots, C_m\}$ of $M$ constraints over a set $\{x_1, x_2, \ldots, x_V\}$ of $V$ variables. $J$ is an upper bound on number of satisfying assignments to any of the constraints and let $B_i$ denote the number of constraints in which variable $x_i$ participates. Let $T = \max_i B_i$.

The reduction will transform Yes instances of the CSP to instances of DirEDPwC where one can find edge-disjoint paths connection at least a fraction $\mathsf{sat}_y$ of the source-destination pairs. If the original instance of the CSP is a No instance, then the in the resulting DirEDPwC instance only a small fraction of source-destination pairs can be routed, even if one allows a large congestion on the edges.

We shall present the reduction starting from a generic CSP, and plug in suitable CSPs that imply strong bounds later in Section 5.4. In addition to the CSP parameters $M, V, J, k, p$, the reduction will use three other integer parameters $Y, Z, L$ which will be defined in Section 5.2.2.

Throughout this section, the notation $[Q]$ for an integer $Q \geqslant 1$ denotes the set $\{1, \ldots, Q\}$. We will typically use $i \in [V]$ for a variable index, $j \in [M]$ for a constraint index, and $q \in [p]$ to refer to a possible value assigned to a variable. For each $q \in [p]$ and for constraint $C_j$ containing $x_i$, let $\Gamma_{ijq}$ be the set of satisfying assignments to $C_j$ that set $x_i$ to $q$ and let $\Gamma_{iq}$ be the set of all pairs $(C_j, \zeta)$ such that $C_j$ contains $x_i$ and $\zeta \in \Gamma_{ijq}$. Note that $\zeta$ is a partial assignment specifying the values for variables that occur in constraint $C_j$.

The construction will be composed of $ZL$ *blobs*, indexed by $(z, l)$ where $z \in [Z]$ and $l \in \mathcal{L}$, where $\mathcal{L}$ is a set of $L$ *labels* that is closed under a certain addition operator. Each blob will consist of $V$ variable gadgets, one for each variable in the CSP. For each variable $x_i$, we give a randomized construction of a gadget called $G_i^{z,l}$ that we describe in detail below. Recall that $B_i \leqslant T$ denotes the number of occurrences of $x_i$ in the constraints. The variable gadget $G_i^{z,l}$ has a matching $M_{z,l}^{(i)}$ consisting of $YJB_i$ *special* edges $e^i_{z,l,s} = (a^i_{z,l,s}, b^i_{z,l,s})$ for $s \in [YJB_i]$ — the vertex $a^i_{z,l,s}$ (resp. $b^i_{z,l,s}$) will be referred to as the left (resp. right) endpoint of the edge $e^i_{z,l,s}$.

In each blob, these disjoint matchings will be strung together by $k$ intermediate levels of *connector vertices* in a random way as described below. Let $\tau = (C_j, \zeta, y)$ be a (constraint,assignment,y) triple where $\zeta$ satisfies $C_j$ and $y \in [Y]$ is arbitrary; we call such a triple an *accepting interaction*. For each $t \in [k+1]$ and each accepting interaction $\tau$, we have a connector vertex $w_\tau^{z,l,t}$. For a variable $x_i$ in $V_t$ and $q \in [p]$, we define a set $W_{z,l,q}^{(i)}$ of these connector vertices as follows: the set $W_{z,l,q}^{(i)}$ consists of all the connector vertices $w^{z,l,t}_{(C_j,\zeta,y)}$ such that $C_j$ uses the variable $x_i$ and the assignment $\zeta$ assigns value $q$ to $x_i$, i.e. the pair $(C_j, \zeta)$ belongs to the set $\Gamma_{iq}$. Thus the cardinality of $W_{z,l,q}^{(i)}$ is $Y|\Gamma_{iq}|$ and thus $\sum_{q\in[p]} |W_{q,z}^{(i)}| \leqslant YJB_i$. Now comes the crucial interconnection of the different matchings via the connector vertices. For each $q \in [p]$, pick **independently and uniformly at random** a subset $S_{z,l,q}^{(i)}$ of the matching $M_{z,l}^{(i)}$ of size $|W_{z,l,q}^{(i)}|$. Connect the left endpoints of the edges in $S_{z,l,q}^{(i)}$ to the vertices $W_{z,l,q}^{(i)}$ via a **random matching**. If the left endpoint of an edge in $S_{z,l,q}^{(i)}$ is connected to the vertex labeled $w_\tau^{z,l,t}$, then the right endpoint of that edge is connected to the corresponding node $w_\tau^{z,l,t+1}$. Moreover, we will call this special edge as $f_\tau^{z,l,i}$. Note that the collection of the edges $f^{z,l,i}_{(C_j,\zeta,y)}$ as $(C_j, \zeta)$ ranges over $\Gamma_{iq}$ and $y$ ranges over $Y$ is precisely the sub-matching $S_{z,l,q}^{(i)}$ of $M_{z,l}^{(i)}$.

This defines a blob for every $z, l$. We now define how to connect the various blobs. Let $A \leqslant MJY$ be the number of accepting interactions, and let $u_1, \ldots, u_A$ be the set of convexly independent vectors from Lemma 2.7. Note that each $u_i \in [A] \times [A^2]$. We let $u_\tau$ denote the vector associated with the accepting interaction $\tau$, under some fixed one-one mapping of the accepting interactions into $[A]$.

Recall our definition of the *label set* $\mathcal{L} = [2AZ] \times [2A^2 Z]$. For each $z \in [Z]$, each label $l \in \mathcal{L}$, and each accepting interaction $\tau$, we connect the connector vertex $w_\tau^{z,l,k+1}$ to the connector vertex $w_\tau^{z+1,l+u_\tau,1}$.

Finally, we add sources $S_{j,y,l}$ and destination $T_{j,y,l}$. For each $j, y, l$ and each accepting interaction $\tau$ involving $C_j$ and $y$, we connect $S_{j,y,l}$ to $w_\tau^{1,l+u_\tau,1}$ and connect $w_\tau^{Z,l+Zu_\tau,k+1}$ to $T_{j,y,l}$. This completes the construction.

Note that the graph on blobs is a layered graph, where each edge goes from a blob in layer $z$ to a blob in layer $(z+1)$.

### 5.2.1 Canonical paths

Let $\sigma = (\tau, l) = (C_j, \zeta, y, l)$ be a tuple where $\tau = (C_j, \zeta, y)$ is an accepting interaction and $l \in \mathcal{L}$ is a label. We call such a tuple a *labeled interaction*. Thus for each labeled interaction $\sigma = (C_j, \zeta, y, l)$, we have a canonical path from $S_{j,y,l}$ to $T_{j,y,l}$ that passes through the blobs $(z, l + zu_\tau)$. We refer to it as $P[\sigma]$ or $P[j, \zeta, y, l]$.

### 5.2.2 The parameters

The reduction above used several parameters such as $Y, Z$ and $L$. The analysis will use some other parameters such as $r, \rho, X$, etc. We now specify how these are picked, in the order of dependence.

Let $\varepsilon > 0$ be a fixed constant, and $n$ denote the asymptotic size parameter. Let $c = c(n)$ be the congestion that we are seeking hardness for, and set $b = c+1$. For some integer $p \geqslant b$, suppose that we start with a $p$-ary $k$-partite CSP with completeness $\mathsf{sat}_y$ and soundness $\mathsf{sat}_n$. Let $M$ and $V$ denote the number of variables and constraints respectively, and let $J$ be an upper bound on the number of satisfying assignments to any constraint. Further, $B_i$ denotes the number of constraints that variable $i$ participates in.

Here is how the rest of the parameters are defined.

$$
\begin{aligned}
r &= (5c^k \mathsf{sat}_n)^{-1} \quad \text{(stands for the inapproximability ratio guaranteed by the reduction when } \mathsf{sat}_y = 1\text{)} \\
Y &= \max\left(1, \left\lceil \tfrac{2(c+1)p^{k-1}}{J} \right\rceil\right) \\
\rho &= \max(4pkJr, (YJ)^{\frac{1}{c+1}}) \\
X_i &= YJB_i \text{ (for } i \in [V]) \\
X &= \max_i X_i \\
A_i &= X_i/\rho \\
Z &= 8Mr\rho^{c+1} \\
\mathcal{L} &= [2MYJZ] \times [2M^2Y^2J^2Z] \\
L &= |\mathcal{L}| = 4(MYJ)^3 Z^2 .
\end{aligned}
$$

Note that $\sum_{i=1}^V B_i = kM$. For now, the reader may find it easier to think of the congestion $c$ as a large constant.

We finish this section by arguing about the completeness of this reduction. The soundness analysis appears in Section 5.3. Finally, in section 5.4, we use this reduction starting with an appropriate CSP, and show that for this choice, we get the desired hardness.

### 5.2.3 Completeness

Suppose the CSP instance $\phi$ has an assignment, say $\eta$, that satisfies $\mathsf{sat}_y M$ of the constraints. Then we claim that at least $\mathsf{sat}_y MYL$ of source-destination pairs can be routed on edge-disjoint paths. We claim that for each $C_j$ that is satisfied by $\eta$, all the pairs $(S_{j,y,l}, T_{j,y,l})$ where $C_j$ is satisfied by $\eta$, $y \in [Y]$ and $l \in \mathcal{L}$ can be routed on edge-disjoint paths. Indeed, let $C_j$ be satisfied by $\eta$ and let $\zeta$ be the projection of $\eta$ to the variables that participate in constraint $C_j$. Then, for each $y \in [Y]$ and each $l \in \mathcal{L}$, use the canonical path $P[j, \zeta, y, l]$ to connect $S_{j,y,l}$ to $T_{j,y,l}$. Since the various $\zeta$'s used are projections of a single satisfying assignment $\eta$, the paths are edge disjoint by construction.

Lemma 5.2 (Completeness). *Suppose that $\phi$ is a Yes instance. Then one can route at least a fraction $\mathsf{sat}_y$ of all* *the $MYL$ source-destination pairs on edge-disjoint paths, i.e., with congestion* 1. *In particular, if the original CSP had perfect completeness* ($\mathsf{sat}_y = 1$) *then* all *the source-destination pairs can be connected via edge-disjoint paths.*

Thus our reduction from CSP to DirEDPwC preserves perfect completeness. We will only apply this reduction with CSPs that are hard even with perfect completeness (this is done in Section 5.4). Therefore, our hardness results for DirEDPwC show that it is impossible to route more than a small fraction of the input pairs with congestion $c$, even if one is promised that all pairs can be routed via edge-disjoint paths.

### 5.2.4 Size of the DirEDPwC instance

LEMMA 5.3 (REDUCTION COMPLEXITY). *The above reduction produces a graph with at most $O(T)$ vertices and edges and runs in time that is polynomial in $T$ where $T = M^7 p^{4k} r^3 k(4pkJr)^{3b}$, assuming that the parameters choices satisfy $J \leqslant 2bp^{k-1} \leqslant (4pkJr)^b$.*

PROOF. The number of source-destination pairs in the produced instance is $MYL$, each of which has at most $J$ canonical paths of length $O(kZ)$ connecting them. Therefore, the total number of edges in the graph is $O(kZJMYL)$. Let us now recall the value of the parameters from Section 5.2.2. Under the assumption $J \leqslant 2bp^{k-1} \leqslant (4pkJr)^b$, we have $\rho = 4pkJr$ and $YJ \leqslant 4bp^{k-1} = O(bp^{k-1})$. We have $Z = 8Mr\rho^b$ and $L = 4M^3(YJ)^3Z^2$. Therefore, the size is $kZJMYL = O(kM^4(YJ)^4Z^3) = O(kM^7(YJ)^4r^3\rho^{3b}) = O(kM^7p^{4k}r^3(4pkJr)^{3b})$. The time complexity of the reduction is clearly polynomial in the size of the graph it produces. □

## 5.3 Soundness of the reduction

We now show that if no assignment satisfies more than a small fraction of the constraints, then it is impossible to route many of the $(S_{j,y,l}, T_{j,y,l})$ paths, even if congestion $c$ is allowed. This part is complicated with several steps and uses several of the ideas developed by Andrews and Zhang [4].

### 5.3.1 All paths are (nearly) canonical

For a blob $(z,l)$, let $z$ be its *layer* and $l$ be its *label*. Consider the graph $G_L$ formed by shrinking each blob $(z,l)$ to a single node. We say a path $P$ passes through blob $(z,l)$ in $G$ if its image in $G_L$ passes through the vertex $(z,l)$. Recall that $G_l$ is a layered graph, with each edge from a node $(z,l)$ going to a node in layer $(z+1)$. The sources connect to layer 1 and all edges entering a destination originate in layer $Z$.

Consider an arbitrary path from $S_{j,y,l}$ to $T_{j,y,l}$. Our choice of labels, and Lemma 2.7, guarantee the following.

LEMMA 5.4. *The set of blobs that a path from $S_{j,y,l}$ to $T_{j,y,l}$ passes through is identical to that of some canonical path $P[j,\zeta,y,l]$, i.e., equals $\{(z, l+zu_\tau) \mid 1 \leqslant z \leqslant Z\}$ where $\tau = (C_j,\zeta,y)$.*

Let $P$ be a path from $S_{j,y,l}$ to $T_{j,y,l}$. The above claim implies that $P$ corresponds to some canonical path $\hat{P}$ and hence to a labeled interaction $\sigma = (C_j,\zeta,y,l)$. For a variable $x_i$ that this constraint participates in, we say that this labeled interaction *highlights* the value $q$ given by $\zeta$ to $x_i$. If the path $P$ is routed, we shall say that the labeled interaction $\sigma$ is routed.

Note that the path $P$ can deviate from the canonical path $\hat{P} = P[\sigma]$ within a blob. Call a path *deviant* in blob $(z,l)$ if it is not canonical within this blob. Also note that the only edges going from blob $(z,l')$ to $(z+1, l'+u_\tau)$ leave the connector vertex $w_\tau^{z,l',k+1}$. Thus $P$ cannot deviate from $\hat{P}$ in its last special edge in any blob. We record this fact as

LEMMA 5.5. *A path $P$ that is deviant in blob $(z,l)$ cannot deviate from the last special edge on the corresponding canonical path in this blob.*

Recall that we wish to show that any routing of $MYL/r$ demands leads to congestion $c+1$ somewhere. We do this by looking at the labeled interactions corresponding to a routing. Consider a set $\mathcal{S}$ of labeled interactions with $|\mathcal{S}|$ at least $MYL/r$. We shall show that with high probability, routing all interactions in $\mathcal{S}$ causes congestion $c+1$ somewhere. We can then use a union bound over all subsets $\mathcal{S}$ to establish the claim. *The next few definitions are with respect to a particular set of routed labeled interactions $\mathcal{S}$.*

DEFINITION 5.6 (HEAVINESS AND LIGHTNESS). *A (variable, value)-pair is said to be* heavy in blob $(z,l)$ *if more than $A = X/\rho$ of the routed labeled interactions highlight it.*

*We say a labeled interaction $\sigma$ is* heavy in blob $(z,l)$ *if for all its variables $x_i$, the value $q$ highlighted by this path is heavy in the blob. We say $\sigma$ is* light in blob $(z,l)$ *if it is not heavy.*

### 5.3.2 Bounding flow on light paths

We first bound the total light flow through any blob.

LEMMA 5.7. *The total number of routed labeled interactions $\sigma$ that are light in a blob $(z,l)$ is at most $MY/4r$.*

PROOF. Consider a particular blob $(z,l)$. For any variable $x_i$, the total number of labeled interactions that can be light because of it is at most $pA_i$, since for each light value $\alpha$ of variable $x_i$, at most $A_i$ unlabeled interactions that highlight value $\alpha$ for $x_i$ can be routed. Thus the total light flow through the blob is at most $\sum_i pA_i$. Recalling that $A_i = YJB_i/\rho$, and that $\sum_i B_i = kM$, the total amount of light flow through is blob is no more than $MY(pkJ/\rho) \leqslant MY/4r$. □

### 5.3.3 Bounding flow on heavy paths

DEFINITION 5.8. *We call a variable* overambiguous in blob $(z,l)$ *if it is heavy for at least $b = c+1$ different values in this blob. A blob $(z,l)$ is called* overambiguous *if some variable in it is overambiguous. A blob that is not overambiguous is called* unambiguous.

We bound the flow on heavy paths as follows. We first use the soundness of the CSP to show that the heavy flow through any unambiguous blob is at most $MY/4r$ (Lemma 5.10). Next we show that each overambiguous blob gives some probability of leading to a congested edge (Lemma 5.14). Finally, a simple counting argument shows that if the total flow routed is at least $MY/r$, then there are many overambiguous blobs (Lemma 5.13). This will imply an upper bound on the probability of there being no congested edge.

### 5.3.4 Unambiguous blobs

LEMMA 5.9. *Suppose that $\phi$ is a no instance. Then for every unambiguous blob, the number of routed labeled interactions that are heavy in it is at most $\frac{MY}{4r}$.*

PROOF. Assume the contrary and suppose that at least $MY/4r$ heavy labeled interactions are routed through an unambiguous blob $(z,l)$. Thus there are at least $M/4r$ constraints $C_j$ such that some labeled interaction corresponding to $C_j$ is heavy in this blob; we will call such a $C_j$ heavy. Now consider the assignment $\zeta$ resulting from picking a random heavy value for each variable in this blob. The assignment $\zeta$ satisfies a particular heavy constraint $C_j$ with probability at least $1/c^k$. Thus the expected number of constraints satisfied by this assignment is at least $M/(4rc^k)$. On the other hand, the soundness of the CSP implies that this can be no more than $M\mathsf{sat}_n$. Thus $r \geqslant (4c^k\mathsf{sat}_n)^{-1}$. This however contradicts the definition of $r$. □

From Lemmas 5.7 and 5.9, we conclude

LEMMA 5.10. *Suppose that $\phi$ is a no instance. Then for any unambiguous blob $(z,l)$ the number of routed labeled interactions that pass through $(z,l)$ is at most $\frac{MY}{2r}$.*

### 5.3.5 Overambiguous blobs

Let $\mathcal{S}$ be a set of unlabeled interaction such that routing $\mathcal{S}$ makes a blob $(z,l)$ overambiguous. We first lower bound the probability that the canonical paths corresponding to $\mathcal{S}$ will cause congestion $b = c+1$ in blob $(z,l)$.

Let $\alpha_1,\ldots,\alpha_b$ be $b$ values for $x_i$ such that $(x_i,\alpha_q)$ is heavy in $(z,l)$. Recall that $X_i = YJB_i$ is the number of special edges in $M^{(i)}_{z,l}$ where $B_i \leqslant T$ is the number of occurrences of variable $x_i$ in the constraints $C_j$, $j \in [M]$. For convenience, we shall omit the subscript $i$ in the rest of this section and use $A$ and $X$ to refer to $A_i$ and $X_i$ respectively. For $q \in [b]$, the heaviness of $(x_i,\alpha_q)$ implies that there is a set $\mathcal{S}_q \subset \mathcal{S}$ of $A$ labeled interactions that highlight value $\alpha_q$ for $x_i$ in $(z,l)$. Thus for each $q$, by construction, the set of special edges in $M^{(i)}_{z,l}$ used by (the canonical paths corresponding to) the labeled interactions in $\mathcal{S}_q$ is a uniformly random subset of size $A$. Thus for a given special edge $e^i_{z,l,s}$, the probability that it gets used by a labeled interaction in $\mathcal{S}_q$, for some $q$, is $(A/X)$. Thus with probability $(A/X)^b$, $e^i_{z,l,s}$ is used by a labeled interaction from each of the sets $\mathcal{S}_q$, and hence suffers congestion $b$.

Intuitively, since the events "edge $e^i_{z,l,s}$ has congestion $b$" are negatively correlated, the probability that none of these events occurs is no larger than what it would be if they were independent. The following lemma formalizes this:

LEMMA 5.11. *Let $\mathcal{S}$ be a set of labeled interactions that make blob $(z,l)$ overambiguous. Then with probability at least $\frac{YJ}{2\rho^b}$, some edge in $(z,l)$ has congestion $(c+1)$.*

PROOF. Let $B^s_q$ denote the event that edge $e^i_{z,l,s}$ gets used by a labeled interaction in $\mathcal{S}_q$. For each $q$, the events $\{B^s_q : s \in [YJB]\}$ are negatively associated (see e.g. [10]). Further, for $q \neq q'$ and any $s,s'$, the events $B^s_q$ and $B^{s'}_{q'}$ are independent. Therefore the events $\{\cap_q B^s_q : s \in [YJB]\}$ are also negatively associated ([10, Prop. 7]). Finally note that the event $\cap_q B^s_q$ is precisely the event that edge $e^i_{z,l,s}$ has congestion $b$. Thus the probability of a congested edge is bounded below by that in the independent case. The latter probability is at most $(1-(A/X)^b)^X = (1-1/\rho^b)^X \leqslant e^{-X/\rho^b} \leqslant e^{-YJ/\rho^b}$. We conclude that with probability at least $(1-e^{-YJ/\rho^b})$, at least one edge get congestion $b$ in a canonical routing of $\mathcal{S}$. In other words, for any give set of labeled interactions $\mathcal{S}$, the canonical paths corresponding to it have a probability $(1-\exp(-YJ/\rho^b))$ of causing congestion $b$ at each overambiguous blob. For our choice of parameters, $YJ/\rho^b \leqslant 1$, so that this probability is at least $\frac{YJ}{2\rho^b}$. □

We note that the above argument only involves the coin tosses for variable $x_i$.

Next we entertain the possibility of deviant paths. For a particular one of these paths $P[\sigma] = P[j,\zeta,y,l']$, we will first bound the probability of there being a deviant path $P$ avoiding $e^i_{z,l,s}$. Recall that our CSP instance was $k$-partite; let $x_i$ belong to part $V_t$. If $t = k$, Lemma 5.5 implies that $P$ cannot avoid $e^i_{z,l,s}$. We thus assume that $t \neq k$. Let us fix the coin tosses for all parts other than $V_k$. Recall that the deviant path $P$ must enter the blob $(z,l)$ at node $w^{z,l,1}_{C_j,\zeta,y}$ and leave the blob using node $w^{z,l,k+1}_{C_j,\zeta,y}$. Consider the set $W_{bad}$ of connector nodes $w^{z,l,k}_{\tau'}$ reachable from $w^{z,l,1}_{C_j,\zeta,y}$ with $\tau' \neq (C_j,\zeta,y)$; since the connector nodes and the left endpoints of the special edges have outdegree one, and the right endpoints have outdegree at most $p$, there are no more than $p^{k-1}$ nodes in $W_{bad}$. For a node $w^{z,l,k}_{\tau'}$ in $W_{bad}$, it has a path to node $w^{z,l,k+1}_{C_j,\zeta,y}$ only if it uses the same matching edge as $\sigma$ for its variable in $V_k$. This happens with probability no more than $1/X_i \leqslant 1/(YJ)$. Taking a union bound over nodes in $W_{bad}$, the probability that there is a deviant path $P$ within this blob is at most $\frac{p^{k-1}}{YJ}$.

Thus amongst the $c+1$ canonical paths that congest edge $e^i_{z,l,s}$, the expected number that can deviate is at most $\frac{(c+1)p^{k-1}}{YJ}$. For our choice of parameters, this expectation is at most a half, and hence with probability at least $1/2$, none of these paths can find a deviation.

Thus with probability at least $\frac{1}{2}(\frac{YJ}{2\rho^b})$, some edge in this blob has congestion $c+1$, for any set of (possibly deviant) paths $\mathcal{P}$ that correspond to $\mathcal{S}$. Moreover, since $x \geqslant 1-e^{-x}$, we have shown that:

LEMMA 5.12. *Let $(z,l)$ be a blob that is overambiguous with respect to a set $\mathcal{S}$ of labeled interactions. Then with probability at least $(1-\exp(-\frac{YJ}{4\rho^b}))$ over the coin tosses in the blob, any routing of $\mathcal{S}$ has an edge with congestion $(c+1)$.*

### 5.3.6 Putting it together

We first use an averaging argument to show that if the total flow is large, there must be several overambiguous blobs. The proof is deferred to the appendix.

LEMMA 5.13. *Let $\mathcal{S}$ be a set of $MYL/r$ labeled interactions and suppose that $\phi$ is a no instance. Then there are at least $ZL/2r$ overambiguous blobs.*

PROOF. Consider a particular layer $z^* \in [Z]$. Because of the layered structure of the graph, each labeled interaction $\sigma \in \mathcal{S}$ must pass through some blob $(z^*,l), l \in \mathcal{L}$. From Lemma 5.10, at most $MYL/2r$ of the $\sigma$'s could be routed through unambiguous blobs. Thus the flow through overambiguous blobs in this layer is at least $MYL/2r$. Since each blob can allow at most $MY$ flow through it, there must be at least $L/2r$ overambiguous blobs in this layer. Moreover, this is true for each $z^* \in [Z]$, and we get a total of $ZL/2r$ overambiguous blobs. □

We next combine Lemmas 5.12 and 5.13 to show the following.

LEMMA 5.14. *Let $\mathcal{S}$ be a set of $MYL/r$ labeled interactions and suppose that $\phi$ is a no instance. Except with probability at most $\exp(-\frac{YJZL}{8r\rho^b})$ over the coin tosses of the reduction, every routing of $\mathcal{S}$ causes congestion $(c+1)$.*

PROOF. By Lemma 5.13, there must be at least $ZL/2r$ overambiguous blobs. By Lemma 5.12, the probability that any one of these blobs avoids congestion $(c+1)$ is at most $\exp(-\frac{YJ}{4\rho^b})$. Since the coin tosses in the blobs are independent, the probability of avoiding congestion everywhere is at most $\exp(-\frac{YJZL}{8r\rho^b})$. □

Taking a union bound over the at most $2^{MYLJ}$ possible sets $\mathcal{S}$, the probability that there is any routing with small congestion is exponentially small for $Z = 8r\rho^b M$. Thus we have shown that:

THEOREM 5.15 (SOUNDNESS). *If $\phi$ is a no instance, then with high probability over the coin tosses of the reduction, it is not possible to route more than $MYL/r$ of the source-destination pairs, even allowing for congestion $c$.*

We conclude this section by noting that Lemmas 5.2, 5.3 and Theorem 5.15, together with the value of the gap $r = \frac{1}{5c^k \mathsf{sat}_n}$, abstract all that we will need about the reduction in the next section.

## 5.4 Using the reduction

In this section, we state and obtain the hardness results for constraint satisfaction problems with certain parameters. These will then be plugged into the above reduction to deduce inapproximability results for routing on directed graphs.

### 5.4.1 Derandomized Serial Repetition

The most obvious way to boost the soundness of an instance $\mathcal{I}$ of a hard CSP with parameters $(M', V', J, p, \ell, 1, s)$ instance $\mathcal{I}$ is to consider the $t$-fold repetition of $\mathcal{I}$ which has $V = V't$ variables ($t$ copies for each of the original variables) and has a constraint for each $t$-tuple of constraints in $\mathcal{I}$, with the $i$'th copy of the variables participating in the $i$'th constraint of each tuple. This yields a CSP with much smaller soundness $s = s'^t$, but unfortunately its large number $M'^t$ of constraints precludes getting meaningful hardness results when $t$ is large.

A more size efficient transformation is to not consider all $t$-tuples but rather consider all tuples corresponding to length $t-1$ walks in a sparse $D$-regular graph $G$ with $M'$ vertices (that correspond to the constraints of $\mathcal{I}$). This will yield a collection of $M'D^{t-1}$ constraints, which is much smaller than $M'^t$ for $D \ll M$. Of course, one cannot argue that the soundness is now as small as $s'^t$. However, if $G$ is a good (spectral) expander, the second largest eigenvalue (in absolute value) of whose adjacency matrix is much smaller than $D$, then one can show that the soundness is not much larger. Since our overall reduction is randomized anyway, we do not even need explicit constructions of expanders. We can pick a random $D$-regular graph (possibly with multiple edges), say by picking $D$ random matchings, with $M'$ vertices, and then appeal to a result that with probability at least $3/4$, the second largest eigenvalue will be at most, say, $2D^{7/8}$ [11]. One can thus show the following.

LEMMA 5.16. *For any integer "repetition parameter" $t \geqslant 1$, there is a randomized reduction that, with probability at least $3/4$, maps instances of a $p$-ary $\ell$-partite CSP with parameters $(M', V', J, p, \ell, 1, s)$ into instances of a $p$-ary $t\ell$-partite CSP with parameters*

$$(M \leqslant M'(4/s)^{4t},\ V = V't,\ J^t,\ p, \ell t, 1, (2s)^t)\ .$$

*The reduction runs in time that is bounded by a polynomial in $M'(4/s')^t$.*

### 5.4.2 Hard CSPs from Raz two prover systems

We start with the following hardness result for Label Cover.

THEOREM 5.17. *There exists an absolute constant $\gamma$, $0 < \gamma < 1$, such that for all large enough integer valued functions $p : \mathbb{N} \to \mathbb{N}$, it is not possible to decide the CSP with parameters $(n^{\log p},\ n^{\log p},\ p,\ p,\ 2,\ 1,\ 1/p^\gamma)$ in time polynomial in the size of the instance, unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{O(\log p(n))})$. (Here we used the shorthand $p = p(n)$.)*

In particular, using the above, one can get quasi NP-hardness for a soundness of $1/2^{O(\log^{1-\varepsilon} M)}$ as a function of the number of constraints for any desired constant $\varepsilon > 0$. One can also get NP-hardness for a arbitrarily small constant soundness. But the above does not yield a polynomially small soundness of $1/M^\varepsilon$ for some $\varepsilon > 0$. For this we boost the soundness using the expander walk technique discussed in Section 5.4.1.

For an integer valued function $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(n) \leqslant \log n$, we start with a hard instance of the above CSP for the choice $p(n) = 5 \cdot (2c(n)^2)^{2/\gamma}$. We then perform the derandomized serial repetition of Lemma 5.16 on such a CSP with repetition parameter $t = \left\lceil \frac{\log n}{c(n)} \right\rceil$ . This proves the hardness of a CSP with parameters

$$\left(M \leqslant n^{\log p}(4p)^{4t},\ V = tn^{\log p},\ J = p^t,\ p,\ k = 2t,\ 1,\ (2/p^\gamma)^t\right)\ . \quad (1)$$

In what follows, we use the shorthand $c = c(n)$. For the above choice, we have $4(pkJr)^b \geqslant 2bp^{k-1}$: indeed $b = c+1 \geqslant 2$, so $J^b \geqslant p^k \geqslant bp^{k-1}$. Moreover, $J = p^{k/2} \leqslant 2bp^{k-1}$. Therefore, by Lemma 5.3, the size (number of vertices) $S$ of the graph produced by the reduction on the resulting CSP satisfies

$$\begin{aligned} S &= O(M^7 p^{4k} kr^3 (4pkJr)^{3b}) \\ &\leqslant n^{O(\log p)} p^{O(kc)} = p^{O(\log n + kc)} \leqslant p^{O(kc)} \end{aligned}$$

where we used the parameter values from (1), in the last step we used the fact that $kc = 2tc \geqslant 2\log n$, and in the first inequality we used the fact that gap $r$ is given by

$$r = \frac{1}{5c^k \mathsf{sat}_n} = \frac{p^{\gamma k/2}}{5c^k 2^{k/2}} \leqslant p^{\gamma k/2}\ .$$

By our choice of $p$ above, we also have $r \geqslant p^{\gamma k/4}$. It follows that as a function of the number of vertices $S$ of the graph, the gap $r$ satisfies $r \geqslant S^{\Omega(\gamma/c)}$. Therefore, we get the desired polynomial hardness as a function of the congestion.

Let us now bound $S$ as a function of $n$ (this also serves as a bound on the running time of the reduction from the CSP to the directed routing instance). We have

$$S \leqslant p^{O(\log n + kc)} \leqslant p^{O(\log n)} = c(n)^{O(\gamma^{-1}\log n)} = n^{O(\gamma^{-1}\log c(n))}.$$

When $c(n) = \log n$, we have $S = n^{O(\log\log n)}$, and so we get a superconstant $S^{\Omega(\gamma/c)}$ factor inapproximability result for congestion $c$ (as a function of $S$) up to $\Omega(\frac{\log S}{\log\log S})$.

The preceding discussion therefore implies the following hardness result for DirEDPwC:

Theorem 5.18. *There exist absolute constants $\alpha_0, \gamma_0 > 0$ such that for every integer-valued function $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(n) \leqslant \log n$ (and computable in time polynomial in $n$), the following holds unless $\mathrm{NP} \subseteq \bigcup_d \mathrm{BPTIME}(n^{d \log c(n)})$. Given a directed graph $G$ on $S \leqslant n^{O(\log c(n))}$ vertices with source-destination pairs $(s_i, t_i)$, $1 \leqslant i \leqslant k$, it is impossible to distinguish between the following cases in time polynomial in $S$:*

- [Yes Instances:] *There are edge-disjoint paths connecting all the $s_i$-$t_i$ pairs.*
- [No Instances:] *For $c = c(n)$, any routing of more than a fraction $1/S^{\gamma_0/c}$ of the $s_i$-$t_i$ pairs uses some edge at least $c+1$ times. That is, with congestion $c$, at most a fraction $S^{-\gamma_0/c}$ of the $s_i$-$t_i$ pairs can be routed.*

Using the choice $c(n) = \log n$, we get a gap of $(\log S)^{\Omega(1)}$ for congestion $c(S) = \Theta\left(\frac{\log S}{\log\log S}\right)$. This implies a factor $\Omega(\log S/\log\log S)$ hardness for the congestion minimization problem, which we record formally below.

Corollary 5.19. [Hardness of congestion minimization] *Assume that $\mathrm{NP} \not\subseteq \bigcup_d \mathrm{BPTIME}(n^{d \log\log n})$. There is an absolute constant $a_0 > 0$ such that given a directed graph on $S$ vertices with source-destination pairs $(s_i, t_i)$, $1 \leqslant i \leqslant k$, it is impossible to distinguish between the following cases in time polynomial in $S$:*

- [Yes Instances:] *There are edge-disjoint paths connecting all the $s_i$-$t_i$ pairs.*
- [No Instances:] *Every routing of all the the $s_i$-$t_i$ pairs incurs congestion more than $\frac{a_0 \log S}{\log\log S}$ on some edge.*

We note that given a target congestion function $c' = c'(S)$, one needs to choose an appropriate function $c(n)$ so that the value $c(n)$ is equal to $c'(S) = c'(c(n)^{O(\gamma^{-1}\log n)})$. For any $c'$ such that $1 \leqslant c'(S) \leqslant O(\gamma \log S/\log\log S)$, this can be done. Indeed for a given $n$, such a $c = c(n)$ can be found using binary search in the interval $[1, \log n]$, since $S$ is monotonically increasing in $c$.

**Unique Paths property.** We now briefly discuss how one can get the unique paths property. This modification is based on two observations: firstly, if we start with a 2-partite CSP, the instance produced has no deviant paths, and thus all paths are canonical. Secondly, there is one canonical path for each satisfying assignment to the constraint. Thus if we split each source-destination pair into at most $J$ source-destination pairs, one for each satisfying assignment, we can assign each canonical path to a unique source-destination pair and get a unique-paths instance.

The completeness proof goes through unchanged, except that we do not have perfect completeness any more. We note that in the original construction described earlier, since there is a single source-destination pair for each constraint, a constraint is either routed or not routed. In the modified instance, we have a source-destination pair for each (constraint, satisfying assignment) pair, and several such pairs corresponding to a single constraint may get routed. However, since any particular satisfying assignment chosen in the proof of Lemma 5.9 can satisfy at most one (constraint, satisfying assignment) pair involving a particular constraint, the probability of a particular constraint being satisfied by an assignment grow linearly with the number of routed heavy labeled interactions involving the constraint and thus the lemma continues to hold. It is easy to verify that the rest of the proof of soundness goes through unchanged.

We next analyze the bound proved by this reduction. Since we need the CSP to be 2-partite, we cannot use serial repetition any more. We start with a CSP from Theorem 5.17, with $p = 2^{\lfloor c(n)^{\varepsilon(n)} \rfloor}$, for $c(n) = \log n$, and a suitably chosen function $\varepsilon : \mathbb{N} \to \Re$. It is easy to check that $S \leqslant 2^{14c^{1+\varepsilon}}$. Moreover $r = \frac{p^\gamma}{10c^2} \geqslant p^{\frac{\gamma}{2}} = S^{\Omega(\gamma/2c)}$, as long as $p^{\frac{\gamma}{2}} \geqslant 10c^2$. Substituting $p = 2^{\lfloor c(n)^{\varepsilon(n)} \rfloor}$, this bound translates to $\varepsilon \geqslant \frac{\log\log\log n + \log\frac{4}{\gamma}}{\log\log n}$. For $\varepsilon$ set to this lower bound, we get an instance size $S = n^{O(\log\log n)}$. This gives a $S^{\Omega(\frac{1}{c})}$ hardness result for congestion $c = \frac{\delta \log S}{\log\log S}$ for a constant $\delta = \delta(\gamma)$, assuming that $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(n^{O(\log\log(n))})$. As we increase $\varepsilon$ to a large enough constant, we get a hardness result for congestion $(\log S)^\beta$ for any desired constant $\beta > 0$ (now assuming $\mathrm{NP} \not\subseteq \mathrm{BPTIME}(n^{polylog(n)})$).

In fact, given a target congestion value $c(S)$ such that $(\log S)^\beta \leqslant S \leqslant \frac{\delta \log S}{\log\log S}$, we can set $n = 2^{c(S)}$; $\varepsilon$ is set by $1 + \varepsilon = \frac{\log((\log S)/14)}{\log c}$. Combined with Theorem 4.8, we we get the $S^{\Omega(\frac{1}{c})}$ hardness with the unique-paths property for all $c$ such that $1 \leqslant c \leqslant \frac{\delta \log S}{\log\log S}$. We record this as:

Theorem 5.20. *There exist absolute constants $\beta_0, \gamma_0, \delta_0 > 0$ such that for every integer-valued function $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(S) \leqslant \frac{\delta_0 \log S}{\log\log S}$ (and computable in time polynomial in $n$), the following holds unless $\mathrm{NP} \subseteq \mathrm{BPTIME}(n^{\log^{\beta_0} n})$. Given a directed graph $G$ on $S$ vertices with source-destination pairs $(s_i, t_i)$, $1 \leqslant i \leqslant k$, such that there is a unique $s_i$-$t_i$ path in $G$, it is impossible to distinguish between the following cases in time polynomial in $S$:*

- [Yes Instances:] *There are edge-disjoint paths connecting $k_1$ of the $s_i$-$t_i$ pairs.*
- [No Instances:] *For $c = c(n)$, any routing of more than $k_1/S^{\gamma_0/c}$ of the $s_i$-$t_i$ pairs uses some edge at least $c+1$ times. That is, with congestion $c$, at most $k_1 S^{-\gamma_0/c}$ of the $s_i$-$t_i$ pairs can be routed.*

*Since we have the unique-paths property, the hardness holds for* ANFwC *as well.*

### 5.4.3 CSPs with better soundness vs. alphabet size trade-off

The previous result has the drawback that it yields a rather poor constant in the exponent of the $S^{\Omega(1/c)}$ inapproximability factor. An inspection of the above calculation reveals that the main source of this weak bound is the large number $J$ of satisfying assignments compared to the inverse of the soundness $1/\mathsf{sat}_n$ for the CSP which is reduced to the DirEDPwC instance. In turn, this is inherited from the Raz two prover system (Theorem 5.17) where the number of satisfying assignments per constraint is $p$ whereas the soundness is $1/p^\gamma$ for a tiny value of $\gamma > 0$.

To improve the constants, we need to start from a CSP with a better relation between $J$ and the soundness. While better trade-offs should be possible by using more sophisticated CSPs (including possibly a variant of the CSPs constructed by Håstad and Khot [17]), we will use the following result.

THEOREM 5.21. *There exists an absolute constant $\lambda < \infty$ such that for all integer valued functions $p : \mathbb{N} \to \mathbb{N}$ that takes prime values, the $p$-ary 3-partite CSP with parameters*

$$\left(n^{\lambda \log p} 2^{p^\lambda},\ n^{\lambda \log p} 2^{p^\lambda},\ 2p^2,\ p,\ 3,\ 1,\ \frac{3}{p}\right)$$

*is not decidable in time polynomial in the size of the instance, unless* $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{p(n)^{O(1)}} n^{O(\log p(n))})$. *(Here we used the shorthand $p = p(n)$.)*

This can be proved using the powerful (and by now standard) paradigm involving suitable tests on Long Code based encodings of answers in the Raz 2-prover system which are then analyzed using Fourier techniques [16]. In particular, the result above can be obtained using $p$-ary Long Codes where $p$ is a prime, and each test checks whether a certain linear combination of three variables equals one of two possible values modulo $p$. We omit the details here.

Note that since the bound on number $M$ of constraints of the instance is at least $2^{p(n)^{O(1)}}$, and $p(n)$ is larger than the congestion parameter $c(n)$, the largest congestion for which we will get a hardness result by this approach the soundness as a function of $M$ is at most $\log^\gamma M$ for some small $\gamma > 0$.

We will now use the above CSP to obtain a hardness factor of $\Omega(S^{\frac{1}{(9+\varepsilon)(c+9)}})$ for DirEDPwC with congestion $c$, for any desired constant $\varepsilon > 0$, that is valid for congestion $c = c(S)$ in the range $1 \leqslant c \leqslant (\log S)^{\alpha(\varepsilon)}$ for some constant $\alpha(\varepsilon) > 0$.

Let $L = L(\varepsilon) > 16/\varepsilon$ be a large enough integer as a function of $\varepsilon$. For an integer valued function $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(n) \leqslant (\log n)^{1/6\lambda L}$, we start with a hard instance of the CSP from Theorem 5.21 choosing $p = p(n)$ to be any prime in the range $[5 \cdot (6c(n)^3)^L, 10 \cdot (6c(n)^3)^L]$. Note that for this choice $p = O(\log^{1/2\lambda} n)$, so that $n^{\lambda \log p} 2^{p^\lambda} \leqslant p^{2\lambda \log n}$.

As we did with the Raz based CSP earlier, we then boost the soundness using the expander-based derandomized serial repetition on such a CSP with repetition parameter

$$t = \left\lceil \frac{2\lambda L \log n}{c(n)} \right\rceil . \qquad (2)$$

This proves the hardness of a CSP with parameters

$$\big(M \leqslant p^{2\lambda \log n} p^{8t}, V \leqslant t p^{2\lambda \log n},\ J = (2p^2)^t,\ p,\ k = 3t,\ 1,\ (6/p)^t \quad\big) .$$

In what follows, we use the shorthand $c = c(n)$. For the above choice, we have $(4pkJr)^b \geqslant 2bp^{k-1}$: indeed $b = c+1 \geqslant 2$, so $J^b = (2p^{2k/3})^b \geqslant p^{4k/3} \geqslant bp^{k-1}$. Hence $\rho = 4pkJr$. Moreover, $J = 2^{\frac{k}{3}} p^{\frac{2k}{3}} \leqslant 2p^{k-1} \leqslant 2bp^{k-1}$. Therefore, by Lemma 5.3, the size (number of vertices) $S$ of the graph produced by the reduction on the resulting CSP satisfies $S = O(M^7 p^{4k} k r^3 (4pkJr)^{3b})$. The gap $r$ is given by $r = (5c^k \mathsf{sat}_n)^{-1} = \frac{p^{k/3}}{5c^k 6^{k/3}}$, so by our choice of $p$, we have

$$p^{(1-1/L)k/3} \leqslant r \leqslant p^{k/3} . \qquad (3)$$

Let us now try and bound $S$ in terms of $r$. To this end, we will bound each of the quantities $J, M$ that figure in the bound for $S = O(M^7 p^{4k} k r^3 (4pkJr)^{3b})$. We have

$$J = (2p^2)^{k/3} = p^{\frac{k}{3}(2 + \frac{1}{\log p})} \leqslant r^{(1+2/L)(2+1/L)} \leqslant r^{2+\varepsilon}$$

using (3), $p \geqslant 2^L$, and $L > 16/\varepsilon$. We have

$$M \leqslant p^{8k/3} p^{2\lambda \log n} \leqslant r^{8(1+2/L)} r^{(1+2/L)c/L} \leqslant r^{8+\varepsilon(c+1)} ,$$

using $L > 16/\varepsilon$.

Moreover, $p^{4k} \leqslant r^{12(1+\frac{2}{L})} \leqslant r^{12+2\varepsilon}$. The remaining term $k(4pk)^{3b}$ in the bound for $S$ is surely no larger than $2^{15kb} \leqslant p^{15kb/\log p} \leqslant p^{2\varepsilon k(c+1)} \leqslant r^{8\varepsilon(c+1)}$. Combining these bounds, we conclude that

$$\begin{aligned} S &= O(M^7 p^{4k} r^3 k (4pk)^{3b} (Jr)^{3b}) \\ &= O\left(r^{7(8+\varepsilon b)} r^{15+2\varepsilon} r^{8\varepsilon b} r^{(3+\varepsilon)3b}\right) \\ &= O\left(r^{9b+71+18\varepsilon b+6\varepsilon}\right) = O\left(r^{(9+18\varepsilon)(b+8)}\right) . \end{aligned}$$

Therefore with $\varepsilon' = 18\varepsilon$, $S = O(r^{(9+\varepsilon')(c+9)})$. We conclude that the gap as a function of the size of the graph is given by $r = \Omega(S^{\frac{1}{(9+\varepsilon')(c+9)}})$.

The running time of the reduction is polynomial in $S$, and thus by the above calucation is $p^{O(kc)} = p^{O(\log n)} = n^{O(\log c(n))}$. By the preceding discussion, we can conclude the following hardness result.

THEOREM 5.22. *For every constant $\varepsilon > 0$, there exists a constant $\beta = \beta(\varepsilon) > 0$ such that for every integer-valued function $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(n) \leqslant \log^\beta n$ (and computable in time polynomial in $n$), the following holds unless* $\mathrm{NP} \subseteq \bigcup_d \mathrm{BPTIME}(n^{d \log c(n)})$. *Given a directed graph $G$ on $S \leqslant n^{O(\log c(n))}$ vertices with source-destination pairs $(s_i, t_i)$, $1 \leqslant i \leqslant k$, it is impossible to distinguish between the following cases in time polynomial in $S$:*

- [YES INSTANCES:] *There are edge-disjoint paths connecting all the $s_i$-$t_i$ pairs.*
- [NO INSTANCES:] *For $c = c(n)$, any routing of more than a fraction $S^{-\frac{1}{(9+\varepsilon)(c+9)}}$ of the $s_i$-$t_i$ pairs uses some edge at least $c+1$ times. That is, with congestion $c$, at most a fraction $S^{-\frac{1}{(9+\varepsilon)(c+9)}}$ of the $s_i$-$t_i$ pairs can be routed.*

# 6. CONCLUSIONS

We showed a factor $N^{\Omega(1/c(N))}$ inapproximability factor for DirEDPwC with congestion paramater $c(N)$ in the range $1 \leqslant c(N) \leqslant \alpha \log N / \log\log N$ for some absolute constant $\alpha > 0$. For constant congestion, we showed a hardness factor of roughly $N^{\frac{1}{3c+O(1)}}$. An obvious open question is whether the hardness factor can be improved to $N^{\frac{1}{(1+\varepsilon)(c+1)}}$, which would essentially match the best known algorithms that achieve an approximation ratio of $O(cN^{1/c})$. Note that for $c = 1$, such a $N^{1/(2+\varepsilon)}$ hardness factor is known [12]. As mentioned in the introduction, even with perfect completeness, one should be able to get a $N^{\frac{1}{(3+\varepsilon)(c+O(1))}}$ hardness factor using our methods by plugging in a better CSP as starting point for our reduction. Reducing the multiplicative factor from $3+\varepsilon$ all the way to $1+\varepsilon$ appears to require additional new ideas, and remains an interesting open question.

Pinning down the approximability of congestion minimization and EDP on *undirected* graphs remains a substantial challenge. For example, for undirected congestion minimization, there is a still a gap between the $O(\log N/\log\log N)$ ratio achieved by randomized rounding and the factor $o(\log\log N)$ inapproximability result of Andrews and Zhang [2]. For undirected EDP, the best known hardness of $\log^{1/2-\varepsilon} N$ [1] is quite far from the polynomial factors achieved by the known approximation algorithms.

**Acknowledgments.** We thank Joel Friedman for useful discussions on the spectral properties of random regular graphs. We would like to thank Chandra Chekuri and Piotr Indyk for many helpful comments.

## 7. REFERENCES


[1] M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 226–244, 2005.

[2] M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 284–293, 2005.

[3] M. Andrews and L. Zhang. Hardness of the undirected edge-disjoint paths problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 276–283, 2005.

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

[5] C. Chekuri and S. Khanna. Edge disjoint paths revisited. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 628–637, 2003.

[6] C. Chekuri, S. Khanna, and F. B. Shepherd. The all-or-nothing multicommodity flow problem. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 156–165, 2004.

[7] C. Chekuri, S. Khanna, and F. B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 183–192, 2005.

[8] J. Chuzhoy and S. Khanna. Hardness of directed routing with congestion. *ECCC Technical Report TR06-109*, August 2006. http://eccc.hpi-web.de/eccc-reports/2006/TR06-109/index.html.

[9] J. Chuzhoy and J. Naor. New hardness results for congestion minimization and machine scheduling. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 28–34, 2004.

[10] D. P. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures and Algorithms*, 13(2):99–124, 1998.

[11] J. Friedman. A note on the second eigenvalue of regular graphs. *Manuscript (personal communication)*, 2006.

[12] 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. *J. Comput. Syst. Sci.*, 67(3):473–496, 2003.

[13] V. Guruswami and K. Talwar. Hardness of low-congestion routing in undirected graphs. *Manuscript*, May 2005.

[14] V. Guruswami and K. Talwar. Hardness of low congestion routing in directed graphs. *ECCC Technical Report TR06-141*, 2006, http://eccc.hpi-web.de/eccc-reports/2006/TR06-141/index.html.

[15] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[16] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[17] J. Håstad and S. Khot. Query efficient PCPs with perfect completeness. *Theory of Computing*, 1(7):119–148, 2005.

[18] S. Khot. Improved inaproximability results for maxclique, chromatic number and approximate graph coloring. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 600–609, 2001.

[19] J. M. Kleinberg. *Approximation algorithms for disjoint paths problems*. PhD thesis, MIT, May 1996.

[20] S. G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proceedings of the 6th International Conference on Integer Programming and Combinatorial Optimization*, pages 153–168, 1998.

[21] P. Raghavan and C. D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.

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

[23] K. R. Varadarajan and G. Venkataraman. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *Proceedings of the ACM-SIAM Symposium on Algorithms*, pages 379–380, 2004.