Electronic Colloquium on Computational Complexity, Report No. 141 (2006)

Hardness of Low Congestion Routing in Directed Graphs

Venkatesan Guruswami*
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195
venkat@cs.washington.edu

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

Abstract

We prove a strong inapproximability result for routing on directed graphs with low congestion. Given as input a directed graph on N vertices and a set of source-destination pairs that can be connected via edge-disjoint paths, we prove that it is hard, assuming NP doesn't have $n^{O(\log\log n)}$ time randomized algorithms, to route even a $1/N^{\Omega(1/c(N))}$ fraction of the pairs, even if we are allowed to use each edge on $c(N)$ paths. 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. An important feature of our result is that it holds with perfect completeness, and shows hardness of low-congestion routing of instances where **all** the input source-destination pairs can be routed on *edge-disjoint paths*. Consequently, our result also implies that it is hard to find a routing of all the source-destination pairs that incurs congestion at most $\alpha \log N/\log\log N$, even if there exists an edge-disjoint (i.e., congestion 1) routing of all the pairs. This shows the optimality, up to constant factors, of the approximation guarantee of the classic Raghavan-Thompson algorithm based on randomized rounding of the fractional multicommodity flow solution.

1 Introduction

We prove that the problem of routing a maximum number of given source-destination pairs in a directed graph with N vertices while incurring congestion at most $c(N)$ on an any edge is hard to approximate within an $N^{\Omega(1/c(N))}$ factor for all integer valued functions $c : \mathbb{N} \to \mathbb{N}$ satisfying $1 \leqslant c(N) \leqslant \alpha \log N/\log\log N$. Here $\alpha > 0$ is an absolute constant, and the hardness holds under the assumption that NP $\not\subseteq$ BPTIME$(n^{O(\log\log n)})$. We will refer to this problem of routing on directed graphs with bounded congestion as DirEDPwC.

An important, desirable feature of our inapproximability result is that it holds with perfect completeness. Specifically, we show that on input a directed routing instance where **all** the source-destination pairs can be routed on *edge-disjoint paths* (i.e., with no congestion), it is still hard to

*Supported by NSF CCF-0343672, a Sloan Research Fellowship, and a Packard Fellowship.

ISSN 1433-8092

route more than a $1/N^{\Omega(1/c(N))}$ fraction of the pairs even if congestion $c(N)$ is allowed. Note that there is also a natural "bicriteria" aspect to this result — it is hard to find a good solution with congestion $c(N)$ even if a (perfect) solution with much smaller congestion 1 exists. Owing to this aspect, our result also implies as a corollary a factor $\Omega(\log N/\log\log N)$ 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 [5]. 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 [20] achieves an approximation ratio of $O(\log N/\log\log N)$ for the congestion minimization problem. We note that 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 [3].

1.1 Prior Work

When the congestion $c = 1$, DirEDPwC is identical to the classic edge-disjoint paths problem. This was shown to be hard to approximate within a $N^{1/2-\varepsilon}$ factor for any $\varepsilon > 0$ [13], and a natural greedy algorithm is known to achieve an approximation ratio of $O(\min\{N^{4/5}, \sqrt{M})$ (where M is the number of edges) [18, 6].

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 [21, 19]. Hardness results for DirEDPwC with congestion $c > 1$ have been a lot more mysterious. The reduction in [13] completely breaks down even when $c = 2$, and till recently no strong hardness results were known for the case of $c > 1$. Spurred by the breakthrough results of Andrews and Zhang on undirected edge-disjoint paths and congestion minimization [4, 3], a factor $(\log N)^{\Omega(1/c)}$ hardness for the above problem on *undirected graphs* has been shown [2, 14]. This hardness also holds for directed graphs. While the above-mentioned work [5] on directed congestion minimization does not address DirEDPwC explicitly, one can deduce a factor $2^{\log^{1/2-\varepsilon} N/c}$ hardness result, for any $\varepsilon > 0$, for DirEDPwC by adapting their proof in a straightforward way.

For directed congestion minimization, Chuzhoy and Naor [8] were the first to obtain a non-trivial inapproximability result – they established a factor $\Omega(\log\log N)$ hardness result. Prior to [8], 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 Reduction Method

Our result is based on a reduction from general constraint satisfaction problems over large domains. Our proof builds upon the beautiful work of Andrews and Zhang [5] where they showed a $\Omega(\log^{1-\varepsilon} N)$ hardness for the congestion minimization problem and uses several of their ideas. A crucial idea in [5] is that of a labeling scheme for grouping vertices which ensures that routes connecting a source-destinations pair cannot deviate from certain intended "canonical" paths. We use a simple, more efficient labeling scheme that is one of the contributors to our quantitative improvements in the inapproximability factor.

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. 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 of Andrews and Zhang [5], 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 $1 \leqslant c(N) \leqslant O(\log N/ \log\log N)$. However, 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 $N^{1/((9+\varepsilon)(c(N)+9))}$ 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 ε. 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 [16] 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 [16] should continue to hold even with the restriction of k-partiteness.

1.3 Related work

We note that results similar to this work appear in [7]. Their result for DirEDPwC for small congestion c was obtained subsequent to an email communication of the statement of our results, and uses a different approach from their result on congestion minimization. This is unavoidable: the reductions for hardness of congestion minimization in [5] and in [7] both start from the Raz 2-prover system, which does not seem to suffice for getting the $n^{\Omega(1/c)}$ hardness for small c. The initial focus of our approach was on getting polynomially large hardness factor for the constant congestion case, and we confirmed the calculations that our reduction as a bonus also yields a $\Omega(\log N/ \log\log N)$ hardness result for congestion minimization only subsequent to the announcement of such a result by Chuzhoy and Khanna.

As mentioned earlier, our reduction achieves perfect completeness and shows hardness of finding a low congestion routing even if all source-destination pairs can be routed in an edge-disjoint manner. The reduction in [7] (for the case of constant congestion) does not have perfect completeness, but instead offers the "unique paths property," i.e., there is a unique path connecting each of the source-destination pairs in the graph. (The problem with perfect completeness and the unique paths property is of course trivially solvable in polynomial time.) The imperfect completeness seems to be inherent to the proof in [7] for the low congestion case, which reduces from the independent set problem.

2 Background and Definitions

2.1 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 2.1 (General constraint satisfaction problem)**.** *An instance of a (promise) constraint satisfaction problem (CSP) with parameters $(M, N, J, p, k, \mathsf{sat}_y, \mathsf{sat}_n)$ where $M, N, J, p, k : \mathbb{N} \to \mathbb{N}$ 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 N 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 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 sat_n of the constraints.*

The parameters sat_y and 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.*

2.2 Convexly Independent vectors

The labeling scheme we use to group vertices uses a collection of vectors with a restricted linear independence property, defined below.

Definition 2.2. *A set of vectors $v_1, \ldots, v_n$ 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 $v_i = \sum_{j=1}^n \alpha_j v_j$, 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 $v_i = e_i$ 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 v_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 v_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.

Lemma 2.3. *For every integer $n \geqslant 1$, the set of vectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^2$ given by $v_i = (i, i^2)$ is convexly independent.*

Proof. Suppose that for some set of non-negative coefficients $\alpha_j \geqslant 0$ satisfying $\sum \alpha_j = 1$, it is the case that $v_i = \sum_j \alpha_j v_j$. Thus $i = \sum_j j\alpha_j$ and $i^2 = \sum_j j^2 \alpha_j$. Rearranging, we get that $(\sum_j \alpha_j j)^2 = \sum_j \alpha_j j^2$. Let a be the vector given by $a_j = \sqrt{\alpha_j}$ and b be given by $b_j = \sqrt{\alpha_j} j$. Then we have $\|a\| = 1$ and the above condition translates to $(a \cdot b)^2 = \|a\|^2 \|b\|^2$. Thus b must be a scalar multiple of a, which implies that at most one of the α_j's can be non-zero. Along with the fact that $a \cdot b = i$, we conclude that $\alpha_i = 1$ and $\alpha_j = 0$ for $j \neq i$. Thus the vectors v_i are convexly independent. $\square$

3 Reduction from CSP to DirEDPwC

We now describe the reduction. We will show how to transform an instance of a CSP with parameters $(M, N, J, k, p, \mathsf{sat}_y, \mathsf{sat}_n)$ to an instance of DirEDPwC.

Thus our starting point is an instance ϕ of a p-ary k-partite CSP with a set $\{C_1, C_2, \ldots, C_m\}$ of M constraints over a set $\{x_1, x_2, \ldots, x_N\}$ of N 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 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. In addition to the CSP parameters M, N, J, k, p, the reduction will use three other integer parameters Y, Z, L which will be defined in Section 3.2.

Throughout the paper, the notation $[Q]$ for an integer $Q \geqslant 1$ denotes the set $\{1, 2, \ldots, Q\}$. We will typically use $i \in [N]$ 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

Γ_{ijq} be the set of satisfying assignments to C_j that set x_i to q and let Γ_{iq} be the set of all pairs (C_j, ζ) such that C_j contains x_i and $\zeta \in \Gamma_{ijq}$. Note that ζ 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 N 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 $e_{z,l,s}^i = (u_{z,l,s}^i, v_{z,l,s}^i)$ for $s \in [YJB_i]$ — the vertex $u_{z,l,s}^i$ (resp. $v_{z,l,s}^i$) will be referred to as the left (resp. right) endpoint of the edge $e_{z,l,s}^i$.

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 ζ 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 τ, 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 $w_{(C_j,\zeta,y)}^{z,l,t}$ such that C_j uses the variable x_i and the assignment ζ assigns value q to x_i, i.e. the pair (C_j, ζ) belongs to the set Γ_{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 $f_{(C_j,\zeta,y)}^{z,l,i}$ as (C_j, ζ) ranges over Γ_{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 $v_1, \dots, v_A$ be the set of convexly independent vectors from Lemma 2.3. Note that each $v_i \in [A] \times [A^2]$. We let v_τ denote the vector associated with the accepting interaction τ, under some fixed one-one mapping of the accepting interactions into $[A]$.

We define the *label set* $\mathcal{L}$ as $\mathcal{L} = [2AZ] \times [2A^2Z]$. The addition operation on $\mathcal{L}$ is defined to be coordinate wise addition modulo 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 $((a + a' - 1) \mod (2AZ) + 1, (b + b' - 1) \mod (2A^2Z) + 1)$.

For each $z \in [Z]$, $l \in \mathcal{L}$ and each accepting interaction τ, we connect the connector vertex $w_\tau^{z,l,k+1}$ to the connector vertex $w_\tau^{z+1,l+v_\tau,1}$.

Finally, we add sources $S_{j,y,l}$ and destination $T_{j,y,l}$. For each j, y, l and each accepting interaction τ involving C_j and y, we connect $S_{j,y,l}$ to $w_\tau^{1,l+v_\tau,1}$ and connect $w_\tau^{Z,l+Zv_\tau,k+1}$ 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 $(z + 1)$.

3.1 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 $(z, l + zv_\tau)$. We refer to it as $P[\sigma]$ or $P[j, \zeta, y, l]$.

3.2 The parameters

The reduction above used several parameters such as Y, Z and L. The analysis will use some other parameters such as r, ρ, 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 sat_y and soundness sat_n. Let M and N 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.

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

Note that $\sum_{i=1}^{N} B_i = kM$. 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 4. Finally, in section 5, we use this reduction starting with an appropriate CSP, and show that for this choice, we get the desired hardness.

3.3 Completeness

Suppose the CSP instance ϕ has an assignment, say η, 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 η, all the pairs $(S_{j,y,l}, T_{j,y,l})$ where C_j is satisfied by η, $y \in [Y]$ and $l \in \mathcal{L}$ can be routed on edge-disjoint paths. Indeed, let C_j be satisfied by η and let ζ

be the projection of η 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 ζ's used are projections of a single satisfying assignment η, the paths are edge disjoint by construction.

Lemma 3.1 (Completeness)**.** *Suppose that ϕ is a Yes instance. Then one can route at least a fraction 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). 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.

3.4 Size of the DirEDPwC instance

Lemma 3.2 (Reduction Complexity)**.** *The above reduction produces a graph with at most $O(T)$ vertices and edges and runs in time 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 3.2. 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, $kZJMYL = O(kM^4(YJ)^4Z^3) = O(kM^7(YJ)^4r^3\rho^{3b}) = O(kM^7p^{4k}r^3(4pkJr)^{3b})$. The time complexity of the reduction is clearly polynomial in the size of the graph it produces. $\square$

4 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 using several of the ideas developed by Andrews and Zhang [5].

4.1 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 guarantees the following:

Lemma 4.1. *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 $\{(z, l + zv_\tau) \mid 1 \leqslant z \leqslant Z\}$ where $\tau = (C_j, \zeta, y)$.*

Proof. Let P be a path from $S_{j,y,l}$ to $T_{j,y,l}$. Since the graph G_l is layered, with Z layers separating the source from the destination, the path P passes through exactly Z blobs. Let h_i be the number of hops in P that go from some blob (z, l') to $(z + 1, l' + v_i)$, where $z \in Z, l' \in \mathcal{L}$. Let (Z, l'') be the last blob that P passes through. Since there is an edge from (Z, l'') to $T_{j,y,l}$, there is some canonical path $P[j, \zeta, y, l]$ that uses blob (Z, l'') and thus $l'' \equiv l + Zv_\tau$ for some accepting interaction τ corresponding to (C_j, y). Also $l'' \equiv l + \sum_i h_i v_i$. Thus $Zv_\tau \equiv \sum_i h_i v_i$. While the above equality is modulo the grid dimensions $2nZ \times 2n^2 Z$, the fact that each $v_i \in [n] \times [n^2]$ implies that it holds over integers as well. Since $\sum_i h_i = Z$, and the v_i's are convexly independent, it must be the case that $h_\tau = Z$ and $h_{\tau'} = 0$ for $\tau' \neq \tau$. The claim follows. □

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 ζ to x_i. If the path P is routed, we shall say that the labeled interaction σ 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 $(z + 1, l' + v_\tau)$ 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 4.2. *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 4.3 (Heaviness and Lightness)**.** *Call a (variable,value)-pair* heavy in blob (z, l) *if more than $A = X/\rho$ of the routed labeled interactions highlight it.*

We say a labeled interaction σ 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 σ is* light in blob (z, l) *if it is not heavy.*

4.2 Bounding flow on light paths

We first bound the total light flow through any blob.

Lemma 4.4. *The total number of routed labeled interactions σ 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 α of variable x_i, at most

A_i unlabeled interactions that highlight value α 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 $MY(pkJ/\rho) \leqslant MY/4r$. □

4.3 Bounding flow on heavy paths

Definition 4.5. *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 4.7). Next we show that each overambiguous blob gives some probability of leading to a congested edge (Lemma 4.10). Finally, a simple counting argument shows that if the total flow routed is at least MY/r, then there are many overambiguous blobs (Lemma 4.9). This will imply an upper bound on the probability of there being no congested edge.

4.4 Unambiguous blobs

Lemma 4.6. *Suppose that ϕ 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 ζ resulting from picking a random heavy value for each variable in this blob. The assignment ζ 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 4.4 and 4.6, we conclude

Lemma 4.7. *Suppose that ϕ 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 $\frac{MY}{2r}$.*

4.5 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, α_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, α_q) implies that there is a set $\mathcal{S}_q \subset \mathcal{S}$ of A labeled interactions that highlight value α_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 latter probability is at most $(1-(A/X)^b)^X = (1-1/\rho^b)^X \leqslant e^{-X/\rho^b} \leqslant e^{-YJ/\rho^b}$. More formally, 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 (proposition 7 in [10]). Finally note that the event $\cap_q B^s_q$ is precisely the event that edge $e^i_{z,l,s}$ has congestion 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 4.2 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 σ 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 4.8. *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 $(c+1)$.*

4.6 Putting it together

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

Lemma 4.9. *Let $\mathcal{S}$ be a set of MYL/r labeled interactions and suppose that ϕ 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 4.7, at most $MYL/2r$ of the σ'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 4.8 and 4.9 to show the following.

Lemma 4.10. *Let $\mathcal{S}$ be a set of MYL/r labeled interactions and suppose that ϕ 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 4.9, there must be at least $ZL/2r$ overambiguous blobs. By Lemma 4.8, 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 4.11 (Soundness)**.** *If ϕ 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 3.1, 3.2 and Theorem 4.11, 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 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.

5.1 Derandomized Serial Repetition

The most obvious way to boost the soundness of an instance $\mathcal{I}$ of a hard CSP with parameters $(M', N', J, p, \ell, 1, s)$ instance $\mathcal{I}$ is to consider the t-fold repetition of $\mathcal{I}$ which has $N' = Nt$ 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 walks[1] 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 "expander", one can show that the soundness is not much larger. This is based on Lemma 5.2 below that asserts a "strong hitting property" of random walks on expanders (versions of this lemma appear in several papers such as [9, 17], see also [1, Chap. 9]). In what follows, we use the following terminology:

Definition 5.1. *An n-vertex graph (possibly with multiple edges) an (n, d, λ)-graph if it is d-regular and its adjacency matrix has eigenvalues $d = \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, and $\max\{\lambda_2, |\lambda_n|\} \leqslant \lambda$.*

Lemma 5.2. *Let $G = (V, E)$ be an (n, d, λ)-graph, and let $t \geqslant 1$ be an integer. Let $S_0, \ldots, S_t$ be arbitrary subsets of V such that $|S_i| \leqslant \alpha n$ for $i = 0, 1, \ldots, t$. Then the fraction of length t walks in G whose i'th vertex belongs to S_{i-1} for each $i = 1, 2, \ldots, t+1$ is at most $\alpha(\alpha + \lambda/d)^t$.*

A couple of remarks on the specifics of the lemma are in order compared to the version in [1]. The bound in [1] is stated for the case when all the sets S_i are the same, but it applies with essentially no change in proof when we have different sets for different steps of the random walk. The bound above is also a little stronger than the one stated in [1], so we give a quick overview of the main steps without details of the calculations. The fraction of such walks can be shown to be at most $\alpha\|P_t A P_{t-1} A \cdots A P_0\|_2$ where A is the normalized adjacency matrix of G, P_i is a diagonal matrix with 1's in the positions corresponding to S_i for $0 \leqslant i \leqslant t$, and the matrix norm is defined by $\|C\|_2 = \sup_{y \neq 0} \|Cy\|_2/\|y\|_2$. Using $P_i^2 = P_i$ and $\|XY\|_2 \leqslant \|X\|_2\|Y\|_2$, we have $\|P_t A P_{t-1} A \cdots A P_0\|_2 \leqslant \prod_{i=0}^{t-1} \|P_{i+1} A P_i\|_2$. One can show that $\|P_{i+1} A P_i\|_2 \leqslant (\alpha + \lambda/d)$, and this yields the claimed bound.

The above shows that (n, d, λ)-graphs with $\lambda \ll d$ are useful for derandomized serial repetition. For example, if $\lambda/d \leqslant \alpha$, then at most $(2\alpha)^{t+1}$ of the length t random walks in G fall inside a particular "bad" set S_i for every step i. While explicit constructions of such graphs are known, since our reduction to DirEDPwC is anyway randomized, we will use the following randomized construction which offers the advantage of a lot of flexibility with the parameters.[2]

[1] The length of a walk is the number of edges in the walk, so a length t walk is a sequence of $t+1$ vertices $v_0, v_1, \ldots, v_t$ where (v_i, v_{i+1}) is an edge for $0 \leqslant i \leqslant t-1$.

[2] The eigenvalue bound stated is worse than the $2\sqrt{d}$ for Ramanujan graphs since we want a result that also applies when d grows with n. For $d = 2^{o(\sqrt{n})}$, Friedman [12] has shown that a $3\sqrt{d}$ upper bound holds with high probability, and for constant d, $\lambda \leqslant 2\sqrt{d-1} + \varepsilon$ with high probability [11]. We use a weaker bound that holds for all d. Using a better bound will only improve the additive $O(1)$ constant in our $N^{1/(9+\varepsilon)(c+O(1))}$ hardness factor.

Proposition 5.3 (Friedman [12]). *For all large enough integers d, n with d even and $d \leqslant n$, the following holds. Consider a random d-regular (multi)-graph G on n-vertices picked as follows: Pick $d/2$ uniformly at random permutations $\pi_1, \ldots, \pi_{d/2}$ on $[n]$. Form a graph with vertex set $[n]$ and edge set $\{(i, \pi_j(i)), (i, \pi_j^{-1}(i)) \mid i \in [n],\ 1 \leqslant j \leqslant d/2\}$. Then, with probability at least $3/4$, G is an $(n, d, 2d^{7/8})$-graph.*

With the expander background in place, we are now ready to state and sketch the proof of our application to CSPs.

Lemma 5.4. *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 ℓ-partite CSP with parameters*

$$(M',\ N',\ J,\ p,\ \ell,\ 1,\ s)$$

into instances of a p-ary $t\ell$-partite CSP with parameters

$$(M \leqslant M'(2/s)^{8t},\ N = N't,\ J^t,\ p, \ell t, 1, (2s)^t)\ .$$

The reduction runs in time bounded by a polynomial in $M'(2/s)^{8t}$.

Proof. The reduction uses a random D-regular graph G on M' vertices where $D = 256\lceil 1/s^8 \rceil$ (so that $2/D^{1/8} \leqslant s$). This is the only randomized part of the reduction. By Proposition 5.3, G is a $(M', D, 4D^{3/4})$-graph. Now on input a ℓ-partite CSP instance $\mathcal{I}'$ on N' variables and M' constraints, form a $t\ell$-partite CSP instance $\mathcal{I}$ on tN' variables (t copies for each variable). There are $M' \cdot D^{t-1}$ constraints in $\mathcal{I}$, one for each t-tuple of the original constraints that correspond to length $t-1$ random walks on G. A constraint of $\mathcal{I}$ is satisfied iff all of the t constituent constraints in its tuple are satisfied. The i'th constraint in a tuple uses the i'th copy of the respective variables (so $\mathcal{I}$ is indeed $t\ell$ partite). The number of satisfying assignments per constraint of $\mathcal{I}$ is clearly at most J^t.

Clearly, if $\mathcal{I}'$ is satisfiable, then giving the same assignment as a variable v to all its t copies satisfies all the constraints of $\mathcal{I}$. This shows that perfect completeness is preserved.

For the soundness analysis, suppose $\mathcal{I}'$ is at most s-satisfiable. Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_t)$ be an arbitrary assignment to the variables of the instance $\mathcal{I}$, where σ_i, $1 \leqslant i \leqslant t$, is the assignment to the set of i'th copies of the variables. For $1 \leqslant i \leqslant t$, let S_{i-1} be the subset of $[M']$ that corresponds to constraints satisfied by σ_i. By the soundness assumption, note that $|S_i|/M' \leqslant s$ for each i, $0 \leqslant i \leqslant t-1$. The fraction of constraints of $\mathcal{I}$ that are satisfied by the assignment σ is precisely the fraction of length $t-1$ random walks in G whose i'th vertex belongs to S_{i-1} for each $i = 1, 2, \ldots, t$. By Lemma 5.2 this fraction is at most $s(s + 2/D^{1/8})^{t-1} \leqslant s(2s)^{t-1} < (2s)^t$. □

5.2 Hard CSPs from Raz two prover systems

The following result is a cornerstone of inapproximability theory and has served as a very useful starting point for numerous reductions It follows by applying the Raz parallel repetition theorem to a standard 2 prover 1 round proof system implied by the PCP theorem, and is often stated as the hardness of a problem called Label Cover.

Theorem 5.5. *There exists an absolute constant γ, $0 < \gamma < 1$, such that for all large enough integer valued functions $p : \mathbb{N} \to \mathbb{N}$, the CSP with parameters $(n^{\log p},\ n^{\log p},\ p,\ p,\ 2,\ 1,\ 1/p^{\gamma})$ is not decidable 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 5.1.

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

$$p(n) = 5 \cdot (2c(n)^2)^{2/\gamma} \ . \tag{1}$$

We then perform the derandomized serial repetition of Lemma 5.4 on such a CSP with repetition parameter

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

This proves the hardness of a CSP with parameters

$$\left(M \leqslant n^{\log p}(2p)^{8t},\ N = tn^{\log p},\ J = p^{t},\ p,\ k = 2t,\ 1,\ (2/p^{\gamma})^{t}\right) \ . \tag{3}$$

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 $J^{b} \geqslant p^{k} \geqslant bp^{k-1}$. Moreover, $J = p^{k/2} \leqslant 2bp^{k-1}$. Therefore, by Lemma 3.2, the size (number of vertices) S of the graph produced by the reduction on the resulting CSP satisfies

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

where we used the parameter values from (3), 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 in (1), 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 $\mathsf{DirEDPwC}$:

Theorem 5.6 (Hardness of DirEDPwC)**.** *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. This slightly improves the $(\log S)^{1-\varepsilon}$ inapproximability shown by Andrews and Zhang [5]. The improvement is due to our choice of label set, which is based on Lemma 2.3 and is much smaller than the choice made in [5], and consequently yields smaller routing instances.

Corollary 5.7 (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.

5.3 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 5.5) 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 [16]), we will use the following result.

Theorem 5.8. *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*

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

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 [15]. 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 $1/\log^\gamma M$ 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 $\mathsf{DirEDPwC}$ 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 ε. 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 5.8 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

$$t = \left\lceil \frac{2\lambda L \log n}{c(n)} \right\rceil . \tag{4}$$

This proves the hardness of a CSP with parameters

$$\big(M \leqslant p^{2\lambda \log n} p^{8t}, N \leqslant t p^{2\lambda \log n},\ J = (2p^2)^t,\ p,\ k = 3t,\ 1,\ (6/p)^t\big) . \tag{5}$$

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 3.2, 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

$$p^{(1-1/L)k/3} \leqslant r \leqslant p^{k/3} . \tag{6}$$

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 (6), $p \geqslant 2^L$, 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 at most $2^{15kb} \leqslant p^{15kb/\log p} \leqslant p^{2\varepsilon k(c+1)} \leqslant r^{8\varepsilon(c+1)}$. Combining these bounds, we conclude that

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

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 calculation 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 5.9. *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 Concluding remarks and an open question

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 absoluate constant $\alpha > 0$. For constant congestion, we showed a hardness factor of roughly $N^{\frac{1}{(9+\varepsilon)c}}$. 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 [13]. As mentioned in the introduction, 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.

Acknowledgments. We thank Joel Friedman for useful discussions on the spectral properties of random regular graphs, and communicating Proposition 5.3 to us.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley and Sons, Inc., 1992.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] J. Chuzhoy and S. Khanna. Hardness of directed routing with congestion. *ECCC Technical Report TR06-109*, 2006.

[8] 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.

[9] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources (extended abstract). In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 14–19, 1989.

[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 proof of Alon's second eigenvalue conjecture and related problems. *CoRR*, cs.DM/0405020, 2004.

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

[13] V. Guruswami, S. Khanna, R. Rajaraman, F. B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *J. Comput. Syst. Sci.*, 67(3):473–496, 2003.

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

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

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

[17] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 248–253, 1989.

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

[19] 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.

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

[21] 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.

ECCC ISSN 1433-8092

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