New Hardness Results for Undirected Edge Disjoint Paths

Julia Chuzhoy* Sanjeev Khanna†

July 2005

Abstract

In the edge-disjoint paths (EDP) problem, we are given a graph G and a set of source-sink pairs in G. The goal is connect as many pairs as possible in an edge-disjoint manner. This problem is NP-hard and the best known approximation algorithm gives an $\tilde{O}(\min\{n^{2/3}, \sqrt{m}\})$-approximation for both directed and undirected graphs; here n and m denote the number of vertices and edges in G respectively. For directed graphs, this result is tight as a function of m since it is known that directed EDP is NP-hard to approximate to within $\Omega(m^{1/2-\epsilon})$ for any $\epsilon > 0$. However, for undirected graphs, until recently nothing better than APX-hardness was known. In a significant improvement, Andrews and Zhang [1] showed that undirected EDP is $\Omega(\log^{1/3-\epsilon} n)$-hard to approximate unless NP is contained in ZPTIME($n^{\mathrm{polylog(n)}}$).

In this paper, we improve the hardness result of [1] as well as obtain the first polylogarithmic integrality gaps and hardness results for undirected EDP when congestion is allowed. A solution to EDP has congestion c if we allow up to c paths to share an edge. When no congestion is allowed, we establish an $\Omega(\log^{1/2-\epsilon} n)$-hardness for EDP. With congestion c, we show that the natural multicommodity flow relaxation of EDP has an $\Omega((\frac{\log n}{(\log\log n)^2})^{1/(c+1)}/c)$ integrality gap. Finally, we show that it is possible to obtain a hardness result that is comparable to the integrality gap. In particular, we show that EDP is $\Omega\left((\log n)^{(1-\epsilon)/(\frac{3}{2}c+\frac{1}{2})}\right)$-hard to approximate for any constant $\epsilon > 0$, when congestion c is allowed, for any $c = o(\log\log n)/(\log\log\log n)^2$, such that $c = 2^z - 1$ for some integer z. We also obtain super-constant hardness when c is as large as $O(\log\log n)/(\log\log\log n)^2$.

Similar results can be obtained for the All-or-Nothing flow problem, a relaxation of EDP in that the unit flow between each routed source-sink pair does not have to be on a single path. Using standard transformations, these results can also be extended to the node-disjoint versions of these problems as well as to the directed setting.

1 Introduction

We study the approximability of the *edge-disjoint paths* (EDP) problem. We are given a graph $G = (V, E)$ and a set $\{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ of pairs of vertices. The objective is to connect as many pairs as possible via edge-disjoint paths. Even highly restricted cases of EDP correspond to well-studied important optimization problems. For instance, EDP on trees of height one is equivalent to the graph matching problem. EDP and its variants also have a host of applications to network routing, resource allocation, and VLSI design. It is then not surprising that EDP is one of the most well-studied problems in combinatorial optimization. In directed graphs, the problem becomes NP-hard even when we are given only two source-sink pairs [15]. In undirected graphs, the seminal work of Robertson and Seymour [29] gives a polynomial time algorithm for any constant number of pairs. These results are suggestive of the inherent differences between undirected and directed versions of the EDP. However, the tractability of undirected EDP with constant number of pairs does not hold once the number of pairs is allowed to grow as a function of the input size. In particular, the problem is NP-hard even on planar graphs [16].

Consequently, much of the recent work on EDP has focused on understanding the polynomial-time approximability of the problem. While constant or poly-logarithmic approximation algorithms are known

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

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

for restricted classes of graphs such as trees, meshes, and expanders [3, 12, 14, 17, 22, 23], the approximability of EDP in general graphs is not well understood. The best approximation algorithm for EDP in directed graphs has a ratio of $\tilde{O}(\min(n^{2/3}, \sqrt{m}))$ [11, 24, 27, 31, 32] where n and m denote the number of vertices and edges respectively in the input graph. For undirected graphs and directed acyclic graphs, this factor improves to an $O(\sqrt{n})$-approximation ratio [9]. In directed graphs, this is matched by an $\Omega(m^{1/2-\epsilon})$-hardness due to Guruswami *et al.* [18]. In contrast, only APX-hardness was known for undirected EDP until recently when Andrews and Zhang [1], in a significant progress, obtained an $\Omega(\log^{1/3-\epsilon} n)$ hardness, assuming NP is not contained in ZPTIME($n^{\mathrm{polylog(n)}}$).

A related problem is the *all-or-nothing* (ANF) flow problem where for each routed pair, it suffices to provide a unit of flow. Thus ANF is a relaxation of EDP. Recent work has shown that in undirected graphs, ANF is $O(\log^2 n)$-approximable [6, 8]. The $\Omega(\log^{1/3-\epsilon} n)$ hardness result in [1] extends to ANF as well.

Overview of Results and Techniques: In this paper, we focus on the approximability of EDP and ANF in undirected graphs. Our first result is an $\Omega(\log^{\frac{1}{2}-\epsilon} n)$ hardness of approximation for undirected edge disjoint paths (for any $\epsilon > 0$). This hardness result also holds for the All-or-Nothing flow problem. Our proof uses the framework of [1]. However, our construction directly works with the PCP characterization of NP due to [30], avoiding the intermediate step taken by [1] of creating an independent set instance. The high-level idea of the hardness reduction, based on framework in [1], is as following. Given an instance ϕ of 3SAT, we construct a graph G_ϕ which contains a sufficiently large collection of edge-disjoint paths for each accepting configuration u of the verifier on ϕ. These paths are referred to as the *canonical paths* of u. The canonical path collections for any two accepting configurations u and v that disagree on some proof bit are made to "randomly intersect" with each other to encode this conflict. The random intersections ensure that the resulting graph has "high girth". The graph G_ϕ serves as the input graph for an EDP instance and the source-sink pairs are simply the end-points of the canonical path collections. The pairs that are routed along canonical paths conflict with high probability whenever the underlying configurations are in conflict with each other. However, these conflicts can be avoided if pairs choose paths that are not canonical. The high girth property ensures that on average, a non-canonical path is much longer than a canonical path and thus consumes much more of the routing capacity of the graph. As a result, whenever ϕ is not satisfiable, with high probability, a much smaller fraction of pairs can be routed in the graph G_ϕ. In particular, we establish the following theorem:

Theorem 1 *Undirected EDP and ANF are* $\Omega(\log^{1/2-\epsilon} n)$*-hard to approximate unless NP is contained in ZPTIME*($n^{\mathrm{polylog(n)}}$).

We next consider the relaxation of the EDP where we allow each edge to be shared by a small number of paths, say, c. We notice that EDP with congestion is usually considered in a bi-criteria setting, where the performance of an algorithm is compared to an optimal solution to EDP with congestion 1 on edges. It is known that the integrality gap of the multicommodity flow relaxation can be bounded by $O(n^{1/c})$ (even in directed graphs) for any constant congestion $c \geq 2$ [4, 5, 27]. The integrality gap reduces to a constant when congestion is allowed to be $O(\log n/ \log\log n)$ [28]. In planar graphs, when congestion 2 is allowed, the integrality has recently shown to be bounded by $O(\log n)$ [7, 8]. On the negative side, it is known that with $c = 1$, the multicommodity flow relaxation has an integrality gap of $\Omega(\sqrt{n})$ even in planar graphs [17]. However, no superconstant lower bounds on the integrality gap of the multicommodity flow relaxation were known so far, even for $c = 2$. We resolve this question by establishing the following:

Theorem 2 *For any congestion* $2 \leq c \leq O(\log\log n/ \log\log\log n)$*, the integrality gap of undirected EDP is* $\Omega((\frac{\log n}{(\log\log n)^2})^{1/(c+1)}/c)$*. For ANF, the integrality gap with congestion c is* $\Omega((\frac{\log n}{(\log\log n)^2})^{1/(c+1)}/c^2)$*. In particular, for* $c = O((\log\log n)/(\log\log\log n)^2)$*, integrality gaps for both problems are superconstant.*

We note that an immediate consequence of Theorem 2 is that for any integer i, the gap between $1/i$-integral multicommodity flow (i.e. each flow path carries an integral multiple of $1/i$ units of flow) and fractional multicommodity flow is super-constant in undirected graphs. To our knowledge, prior to our work, it was not known if there was a superconstant gap even between half-integral flow and fractional flow in directed or undirected graphs. The instances used in establishing the integrality gap have a surprisingly simple structure.

Our final result shows hardness of approximation bounds similar to above integrality gaps.

Theorem 3 *For any $\epsilon > 0$ and congestion $c = 2^z - 1 \leq o(\log\log n/(\log\log\log n)^2)$ for some positive integer z, undirected EDP and ANF are $\Omega\left(\log^{(1-\epsilon)/(\frac{3}{2}c+\frac{1}{2})} n\right)$-hard to approximate unless NP is contained in ZPTIME($n^{\text{polylog(n)}}$). Moreover, EDP with congestion c is hard to approximate up to some super-constant factor, even for $c = O(\log\log n/(\log\log\log n)^2)$.*

We note that standard transformations allow us to get matching results for the corresponding node-disjoint version of these problems (an undirected edge-problem can be reduced to an undirected node-problem by working on the line graph of the edge-problem).

We have very recently learnt that two other groups [2, 19] have independently discovered results that are similar to Theorem 2 and Theorem 3 above.

Organization: We start by reviewing the parameters of Samorodnitsky-Trevisan PCP construction in Section 2. In Section 3, we establish Theorem 1. Section 4 presents the family of instances that establish Theorem 2. Finally, we establish Theorem 3 in Section 5.

2 Starting Point: A PCP Result

Our starting point is a PCP characterization of NP, proved by Samorodnitsky and Trevisan in [30]. We briefly summarize the construction here; more details can be found in the appendix. Let ϕ be an instance of 3SAT on n variables. For any constant $k > 0$, the ST construction gives a PCP verifier that uses $r = O(\log n)$ random bits to generate $q = k^2$ locations to probe in the proof. The verifier reads these q bits in the given proof Π and decides whether or not ϕ is satisfiable. Given a random string r of the verifier, let $b_1(r), \ldots, b_q(r)$ be the indices of the proof bits read. A *configuration* is $(r, a_1, \ldots, a_q)$, where $a_1, \ldots, a_q \in \{0, 1\}$ are values of $\Pi_{b_1(r)}, \ldots, \Pi_{b_q(r)}$. We say that a configuration $(r, a_1, \ldots, a_q)$ is *accepting*, if, for a random string r of the verifier and the values $a_1, \ldots, a_q$ of proof bits $\Pi_{b_1(r)}, \ldots, \Pi_{b_q(r)}$, the verifier accepts. If ϕ is a Yes-Instance (i.e., ϕ is satisfiable), there exists a proof Π such that the probability that the verifier accepts is at least 1/2. Otherwise, if ϕ is a No-Instance (i.e., it is non-satisfiable), for all proofs Π, the verifier accepts with probability at most 2^{-k^2}. Abusing the notation, we will denote by r both the random string of the verifier and the number of random bits (i.e., the length of the string).

For our reductions, we would assume that this protocol is independently repeated $\lambda = \frac{2\beta\log\log n}{k^2} = O(\log\log n)$ times where $\beta >> k^2$ is a large constant. The verifier now accepts iff the original verifier accepts in each protocol repetition. The resulting PCP has the following properties:

- **Random Bits:** $\lambda r = O(\log n \log\log n)$. Let R denote the set of all possible random string, $|R| = 2^{\lambda r}$.
- **Query Bits:** $q = \lambda k^2 = O(\log\log n)$. W.l.o.g., assume that the verifier reads exactly q bits of proof for every random string.
- **Completeness:** Yes-Instance is accepted with probability at least $2^{-\lambda}$.
- **Soundness:** No-Instance is accepted with probability at most $2^{-\lambda k^2}$.
- For each random string, there are $2^{\lambda(2k-1)}$ accepting configurations.
- For every random string r, for every $j : 1 \leq j \leq q$, the number of accepting configurations where the value of $\Pi_{b_j(r)} = 0$ equals the number of accepting configurations where $\Pi_{b_j(r)} = 1$.
- For each proof bit Π_j let Z_j be the set of all the accepting configurations in which bit Π_j participates with value 0, and let O_j be the set of all the accepting configurations in which Π_j participates with value 1. We denote $n_j = |Z_j| = |O_j|$. Then $n_j \geq 2^{\lambda r/2}$.

Let $\mathcal{C}$ denote the set of all the accepting configurations, $|\mathcal{C}| \leq 2^{\lambda r} \cdot 2^{2\lambda k}$.

3 Hardness of Approximating EDP

In this section, we will establish Theorem 1, namely, EDP is hard to approximate to within a factor of $(\log n)^{1/2-\epsilon}$ for any $\epsilon > 0$. The construction used in this section will also serve as a building block for establishing Theorem 3. The starting point of our reduction is a PCP verifier for 3SAT as summarized in the preceding section. Let ϕ be an instance of 3SAT on n variables. Consider a PCP verifier V for ϕ as described in the preceding section. We will use V to construct an EDP instance on a graph G_ϕ such that if ϕ is satisfiable, at least P_{YI} pairs can be routed, and if ϕ is unsatisfiable, with high probability, only $P_{YI}/\log^{1/2-\epsilon} N$ pairs can be routed; here $N = n^{\text{polylog(n)}}$ denotes the size of G_ϕ. Recall that k is a large constant, and $\lambda = \frac{2\beta \log\log n}{k^2}$. The gap between the yes and the no instances in the PCP construction is close to $2^{\lambda k^2} = \log^{2\beta} n$. In our construction, we will try to make the gap between the yes and the no instances close to $2^{\lambda k^2}$, while the graph size N will be close to $2^{2^{2\lambda k^2}}$, thus proving $\Omega\left(\log^{\frac{1}{2}-\epsilon} N\right)$ hardness of approximation.

We construct our graph in two steps. First, we construct, for each proof bit Π_i, a gadget denoted by $G(i)$. In the second step, we create the final graph, by connecting all the gadgets representing the proof bits, and by adding source and sink pairs.

3.1 Bit Gadget

We will use two parameters M and X to describe the gadget. Consider some proof bit Π_i. We now show how to construct a corresponding gadget $G(i)$. Recall that Z_i, O_i are the collection of all the accepting configurations, in which the value of Π_i is 0 or 1, respectively, with $|Z_i| = |O_i| = n_i$. For each configuration $\alpha \in Z_i \cup O_i$, for each $m : 1 \le m \le M+1$, there are X vertices $v_{x,m}(\alpha, i)$, for $1 \le x \le X$, called *level m vertices*, representing this configuration.

Additionally, for each $m : 1 \le m \le M$, we have Xn_i edges, called *special edges at level m*, and denoted by $(\ell_{a,m}, r_{a,m})$, $1 \le a \le Xn_i$. We also denote the set of left endpoints of these edges by $L_m(i) = \{\ell_{a,m}\}_{a=1}^{Xn_i}$, and the set of right endpoints of these edges by $R_m(i) = \{r_{a,m}\}_{a=1}^{Xn_i}$.

Finally, we show how to connect the vertices representing the configurations with the special edges. This is done by the means of *regular edges*, as follows. Consider level m vertices, for $1 \le m \le M$. We have Xn_i level m vertices, representing configurations in Z_i (denote this set of vertices by $Z_m(i)$), and Xn_i level m vertices, representing configurations in O_i (these vertices are denoted by $O_m(i)$). We perform a random matching between $Z_m(i)$ and $L_m(i)$, and also we perform a random matching between $O_m(i)$ and $L_m(i)$. Additionally, for each $m : 2 \le m \le M+1$, we perform random matchings between $Z_m(i)$ and R_{m-1}, and between $O_m(i)$ and R_{m-1}. The edges participating in these matchings are added to the gadget as regular edges (see Figure 1).

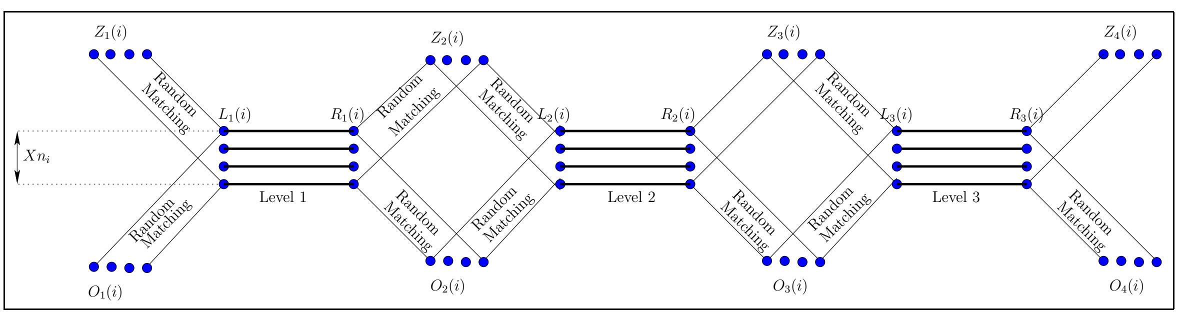

Figure 1: Gadget representing proof bit Π_i for $M = 3$

This concludes the definition of bit gadget. We now define, for each configuration $\alpha \in Z_i \cup O_i$, a collection of X edge-disjoint paths, called *canonical paths*, representing α in gadget $G(i)$. A canonical path $P_x(\alpha, i)$, for $1 \le x \le X$, is defined as: $(v_{x,1}(\alpha,i), \ell_{a_1,1}(i), r_{a_1,1}(i), v_{x_1,2}(\alpha,i), \ldots,$
$\ell_{a_M,M}(i), r_{a_M,M}(i), v_{x_M,M+1}(\alpha,i))$. The indices x_m, a_m for $1 \le m \le M$ are determined by the corresponding

matchings. Therefore, we have X edge-disjoint paths representing α in gadget G_i. Moreover, for all the configurations in Z_i, their Xn_i canonical paths are edge disjoint. The same is true for all the configurations in O_i.

Let $1 \leq m \leq M$, and consider the collection of special edges at level m. Each such edge participates in exactly one canonical path representing a configuration in Z_i, and exactly one canonical path representing a configuration in O_i. Thus, the set of special level m edges defines a random matching between the paths representing the configurations in Z_i and the paths representing the configurations in O_i. In total, gadget G_i defines M random matchings (one matching for each level) between these two sets of paths, and these random matchings are completely independent. Observe that the length of each canonical path is $3M$, and the degree of every vertex is at most 3.

3.2 Bit Gadget Analysis

Set $\Delta = \frac{M}{8 \log M}$, so that $M \geq 8\Delta \log \Delta$ holds. Consider the gadget representing some proof bit Π_i. Let $\mathcal{P}_0$ be the set of canonical paths representing configurations in Z_i, and let $\mathcal{P}_1$ be the set of canonical paths representing configurations in O_i. Recall that $|\mathcal{P}_0| = |\mathcal{P}_1| = Xn_i$.

We say that the gadget is *bad* if there is a pair of subsets $A \subseteq \mathcal{P}_0$, $B \subseteq \mathcal{P}_1$, where $|A| = |B| = \frac{Xn_i}{\Delta}$, such that all the paths in $A \cup B$ are edge disjoint. We say that bad event $\mathcal{B}_1$ happens, if at least one of the gadgets is bad. The proof of the lemma below is similar to a lemma in [1].

Lemma 1 *The probability that gadget $G(i)$ is bad is at most e^{-n}.*

Proof: Consider some proof bit Π_i and its corresponding gadget, and denote $n' = Xn_i$. Let A, B be subsets of $\mathcal{P}_0, \mathcal{P}_1$ respectively of desired sizes. The total number of random matchings between $\mathcal{P}_0$ and $\mathcal{P}_1$ is $n'!$, while the number of matchings with no edge between A and B is:

$$\left(n' - \frac{n'}{\Delta}\right)\left(n' - \frac{n'}{\Delta} - 1\right)\cdots\left(n' - \frac{2n'}{\Delta} + 1\right)\left(n' - \frac{n'}{\Delta}\right)!$$

Therefore, the probability that a random matching does not contain an edge between A and B is:

$$\frac{\left(n' - \frac{n'}{\Delta}\right)\left(n' - \frac{n'}{\Delta} - 1\right)\cdots\left(n' - \frac{2n'}{\Delta} + 1\right)\left(n' - \frac{n'}{\Delta}\right)!}{n'!} \leq \left(1 - \frac{1}{\Delta}\right)^{\frac{n'}{\Delta}} = e^{-\frac{n'}{\Delta^2}}$$

Recall that our gadget defines M matchings between $\mathcal{P}_0$ and $\mathcal{P}_1$, and the gadget is bad with respect to A and B iff none of these matchings contains an edge between A and B.

The probability that none of the M matchings contains an edge between A and B is at most $e^{-\frac{Mn'}{\Delta^2}}$. Finally, the number of possible choices of A and B is:

$$\left[\binom{n'}{\frac{n'}{\Delta}}\right]^2 \leq \left(\frac{n'e}{\frac{n'}{\Delta}}\right)^{2n'/\Delta} = (\Delta e)^{2n'/\Delta}$$

Recall that $M \geq 8\Delta \log \Delta$, and thus the probability that the bipartite graph is bad is at most:

$$e^{-\frac{Mn'}{\Delta^2}} \cdot (\Delta e)^{2n'/\Delta} \leq e^{-\frac{Mn'}{\Delta^2} + \frac{2n' \log \Delta}{\Delta}} \leq e^{-\frac{2n' \log \Delta}{\Delta}}$$

As $n' = n_i X \geq 2^{\lambda r/2} X$, and $\Delta << X$, this probability is less than $e^{-2^{\lambda r/2}} \leq e^{-n}$. □

Corollary 1 *The probability that bad event $\mathcal{B}_1$ happens is at most $\frac{1}{\text{poly}(n)}$.*

Proof: The total number of proof bits is at most $2^r k^2$, and thus using the union bound, the probability that $\mathcal{B}_1$ happens is less than $2^r k^2 e^{-n}$, which is less than $\frac{1}{\text{poly}\, n}$. □

3.3 The Final Instance

Let α be some accepting configuration, and let $i_1, i_2, \ldots, i_q$ be the indices of proof bits participating in α, with $q \leq \lambda k^2$. Consider bit gadget $G(i_j)$, for some $1 \leq j \leq q$. There are X level 1 vertices representing α in $Z_1(i_j) \cup O_1(i_j)$, denote them by $V_j = \{v_{1,1}(\alpha, i_j), \ldots, v_{X,1}(\alpha, i_j)\}$. There are also X level $M+1$ vertices representing α in $Z_{M+1}(i_j) \cup O_{M+1}(i_j)$, denote them by $U_j = \{v_{1,M+1}(\alpha, i_j), \ldots, v_{X,M+1}(\alpha, i_j)\}$.

We add a set of X source vertices representing configuration α, $S(\alpha) = \{s_1(\alpha), \ldots, s_X(\alpha)\}$, and X destination vertices $T(\alpha) = \{t_1(\alpha), \ldots, t_X(\alpha)\}$ (we show how to divide them into pairs later).

We perform a random matching between $S(\alpha)$ and V_1, and also a random matching between $T(\alpha)$ and U_q. Additionally, for each $j : 1 \leq j < q$, we perform a random matching between U_j and V_{j+1}. All the edges in the random matchings are added to the graph as regular edges.

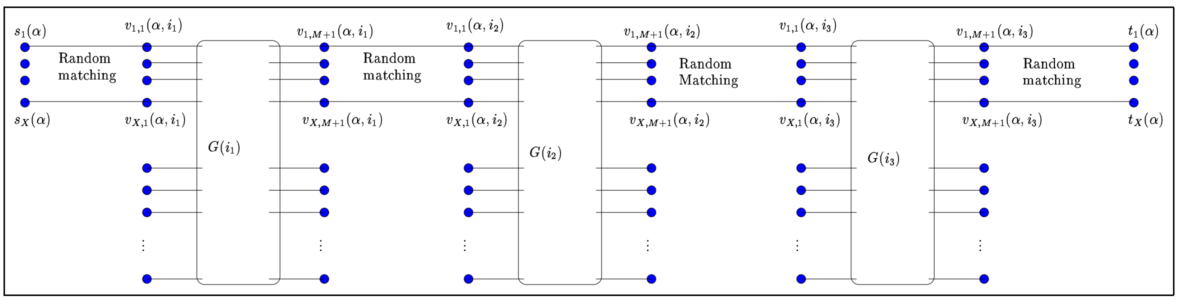

Figure 2: Source-sink pairs for a configuration α.

For each configuration α, we define X canonical paths $P_x(\alpha)$, $1 \leq x \leq X$, representing α, as follows. Let $i_1, \ldots, i_q$ be the indices of the proof bits participating in α, $q \leq \lambda k^2$. For each $x : 1 \leq x \leq X$, we have $P_x(\alpha) = (s_x(\alpha), P_{x_1}(\alpha, j_1), \ldots, P_{x_q}(\alpha, j_q), t_{x'})$, where $x_1, \ldots, x_q, x'$ are determined according to the corresponding matchings.

The graph has the following properties: (i) the length of a canonical path is at most $4M\lambda k^2$; (ii) for each configuration α, there are X canonical paths representing α, all of them edge disjoint; and (iii) the degree of each vertex is at most 3.

Graph size: observe that each graph vertex and edge participate in at least one canonical path. The number of canonical paths is: $X \cdot 2^{\lambda r} \cdot 2^{\lambda(2k-1)}$, and the length of each canonical path is at most $4M\lambda k^2$. It remains to specify the values of the parameters M and X. We will use $M = 2^{\lambda(k^2+k)} = \text{poly}\log n$, and $X = 2^{2^{2\lambda(k^2+4k)}} = 2^{\text{poly}\log n}$. Therefore, the size of the graph is bounded by: $N \leq X \cdot 2^{\lambda r} \cdot M \cdot 2^{2\lambda k}$ $\leq X \cdot 2^{O(\log n \log\log n)} \cdot 2^{O(\log\log n)} \cdot 2^{O(\log\log n)} \leq X \cdot 2^{O(\log n \log\log n)}$.

The source-sink pairs are defined as follows: For each accepting configuration α, the canonical paths $P_1(\alpha), \ldots, P_X(\alpha)$ define a matching between the sources and the destinations corresponding to α. We use this matching to define the source-destination pairs.

Let $\mathcal{P}$ denote the set of all the canonical paths.

3.4 Yes Instance: ϕ is Satisfiable

In the Yes-Instance , there is a PCP proof, for which the acceptance probability is at least $2^{-\lambda}$. For each random string r satisfied by this proof, we can choose all the canonical paths representing the corresponding accepting configuration. All the paths thus chosen are edge disjoint, and the number of chosen paths is at least $P_{YI} \geq X 2^{\lambda r - \lambda} \geq \frac{|\mathcal{P}|}{2^{2\lambda k} \cdot 2^{\lambda}}$. The latter inequality follows from the fact that for each random string r there are at most $2^{2\lambda k}$ accepting configurations.

3.5 No Instance: ϕ is Unsatisfiable

Suppose we have a no instance, and a collection $\mathcal{P}'$ of edge disjoint source-sink paths. We will show that $|\mathcal{P}'|$ can roughly be bounded by $\frac{|\mathcal{P}|}{2^{\lambda k^2}}$. We partition $\mathcal{P}'$ into three subsets, as follows. Let $g = 2^{2\lambda(k^2+k)}$. A non-canonical path is called *long* if its length is more than g. Otherwise, it is called *short*. Let $\mathcal{P}_1 \subseteq \mathcal{P}'$ be the subset of canonical paths, $\mathcal{P}_2, \mathcal{P}_3 \subseteq \mathcal{P}'$ be the subsets of long and short non-canonical paths, respectively. We bound the size of each subset separately.

3.5.1 Canonical Paths

Assume B_1 does not happen. Then in each bit gadget G_i, either the number of paths representing Z_i is less than $n_i X/\Delta$, or the number of paths representing O_i is less than $n_i X/\Delta$. Therefore, if we remove at most $\sum_i n_i X/\Delta$ paths from $\mathcal{P}'$, we obtain a new collection $\mathcal{P}'_1$ of canonical paths, such that in each gadget $G(i)$, we have only paths from Z_i or only paths in O_i. We can thus define a PCP proof as follows: the value of bit Π_i is 0 iff paths representing Z_i are present in $\mathcal{P}'_1$, and it is 1 otherwise. Since we are in a No-Instance , and there are X paths representing each configuration, $|\mathcal{P}'_1| \leq X \cdot 2^{\lambda r}/2^{\lambda k^2}$, which is at most $P_{YI}/2^{\lambda k^2-\lambda}$.

On the other hand, $\sum_i n_i$ can be bounded by $|\mathcal{C}|q \leq 2^{\lambda r+2\lambda k}\lambda k^2$. Also, recall that $\Delta = \frac{M}{8\log M} = \frac{2^{\lambda(k^2+k)}}{8\lambda(k^2+k)}$. Thus $\sum_i \frac{Xn_i}{\Delta} \leq \frac{X2^{\lambda r+2\lambda k}\lambda k^2}{\Delta} \leq \frac{X2^{\lambda r}}{2^{\lambda k^2-2\lambda k}}$ when k is sufficiently large. We can now bound $|\mathcal{P}_1 \setminus \mathcal{P}'_1| \leq \frac{X2^{\lambda r}}{2^{\lambda k^2-2\lambda k}} \leq P_{YI}/2^{\lambda k^2-2\lambda k-\lambda}$. Summing up, $|\mathcal{P}_1| \leq 2P_{YI}/2^{\lambda k^2-2\lambda k-\lambda}$.

3.5.2 Long Non-Canonical Paths

The length of a non-canonical path is at least g. The total number of edges in our graph is at most $|\mathcal{P}| \cdot 4M\lambda k^2$. Therefore, the size of $\mathcal{P}_2$ is bounded by $\frac{|\mathcal{P}|\cdot 4M\cdot\lambda k^2}{g}$. We will show that $\frac{g}{4M\cdot\lambda k^2} \geq 2^{\lambda k^2}$. Recall that $g = 2^{2\lambda(k^2+k)}$, while $M = 2^{\lambda(k^2+k)}$, and thus $4M\lambda k^2 \cdot 2^{\lambda k^2} \leq 4\lambda k^2 \cdot 2^{2\lambda k^2+k\lambda} \leq 2^{2\lambda(k^2+k)} \leq g$. So $|\mathcal{P}_2| \leq \frac{|\mathcal{P}|}{2^{\lambda k^2}} \leq \frac{P_{YI}}{2^{\lambda k^2-2\lambda k-\lambda}}$.

3.5.3 Short Non-Canonical Paths

Suppose there is a short non-canonical path $P \in \mathcal{P}_3$ connecting some source and destination pair (s,t). This path must form a cycle of length at most $g + 4M\lambda k^2 \leq 2g$ with the canonical $s-t$ path. Moreover, at least one edge on the cycle participates in P. Let K denote the number of cycles of length at most $2g$ in our graph. Then $|\mathcal{P}_3| \leq 2g \cdot K$. Our goal is to show that with high probability, K is small. The proof of the claim below is similar a claim in [1].

Lemma 2 *With probability at least $\frac{2}{3}$, $K \leq 2^{4\lambda rg}$.*

Proof: We build a new graph G', obtained by shrinking special edges (each special edge becomes a vertex). Let K' denote the number of cycles of length at most $2g$ in G'. Clearly, $K \leq K'$. We now bound K'.

The probability that some edge e exists in graph G' is at most $\frac{1}{X}$, and the probability that e exists given the existence of i other edges, $1 \leq i \leq 2g$ is at most $\frac{1}{X-2g} \leq \frac{2}{X}$. Therefore, the probability that a given potential cycle that contains g' edges, $g' : 1 \leq g' \leq 2g$ exists is at most $\left(\frac{2}{X}\right)^{g'}$. The number of potential cycles of length g' is bounded by $N^{g'}$. Thus the expected number of cycles of length g' in G' is at most $\left(\frac{2N}{X}\right)^{g'}$. Summing up over all values of g', the expected number of cycles of length at most $2g$ is at most $\left(\frac{2N}{X}\right)^{2g+1}$.

Recall that $N \leq X \cdot 2^{\lambda r+2\lambda k+\lambda(k^2+k)}$, giving the following bound on the expected number of cycles of length at most $2g$:

$$E(K') \leq 2^{\lambda(r+k^2+4k)(2g+1)} \leq 2^{3\lambda rg}$$

Using Markov's inequality, the probability that the number of cycles of length at most $2g$ is greater than $2^{4\lambda rg} \geq 3E(K')$ is at most $\frac{1}{3}$. □

We say that the bad event $\mathcal{B}_2$ happens if $K > 2^{4\lambda rg}$. Assuming B_2 does not happen, we get:

$$|\mathcal{P}_3| \leq 2g \cdot 2^{4\lambda rg} \leq 2^{5\lambda rg} = 2^{5\lambda r\cdot 2^{2\lambda(k^2+k)}} \leq 2^{2^{2\lambda(k^2+3k)+\log\log n}}$$

(since $r = O(\log n)$). Recall that $\lambda = \beta \log\log n/k^2$ for very large constant $\beta >> k^2$, and thus we can assume that $\lambda k \geq \log\log n$, and $|\mathcal{P}_3| \leq 2^{2^{2\lambda(k^2+4k)}} \leq X \leq P_{YI}/2^{\lambda k^2}$.

3.6 Putting it Together

If the events $\mathcal{B}_1$ and $\mathcal{B}_2$ do not happen, then $|\mathcal{P}'| = |\mathcal{P}_1| + |\mathcal{P}_2| + |\mathcal{P}_3| \leq P_{YI}/2^{\lambda(k^2-3k)}$, and thus the gap is $\Omega(2^{\lambda(k^2-3k)})$. Recall that $N = X \cdot 2^{O(\log n \log\log n)} = 2^{2^{2\lambda(k^2+4k)}+O(\log n \log\log n)}$, and so $\log N \leq 2^{2\lambda(k^2+4k)} + O(\log n \log\log n)$. Since $2^{2\lambda(k^2+4k)} > \log^\beta n$, we have that $\log N \leq 2^{2\lambda(k^2+5k)}$, and $\sqrt{\log N} \leq 2^{\lambda(k^2-3k)} \cdot 2^{8\lambda k} \leq \left(2^{\lambda(k^2-3k)}\right)^{1+\frac{8}{k-3}}$. Therefore, the gap is $\log^{\frac{1}{2}-\epsilon} N$, where ϵ is a constant that depends on k and can be made arbitrarily small by choosing k to be sufficiently large.

Now suppose at least one of the events $\mathcal{B}_1$ or $\mathcal{B}_2$ does happen. Then $|\mathcal{P}'|$ may be much larger than the above bound even though ϕ is not satisfiable. But the probability of $B_1 \cup B_2$ is at most $1/\operatorname{poly} n + 1/3 \leq 1/2$. Thus a $\log^{\frac{1}{2}-\epsilon} N$-approximation algorithm for EDP would give us a co-RPTIME($n^{\text{polylog(n)}}$) algorithm for 3SAT. Since 3SAT is in NP, we can use a standard result to convert this into a ZPTIME($n^{\text{polylog(n)}}$) algorithm for 3SAT, giving us our main result.

4 Integrality Gap of the Multicommodity Flow Relaxation

We will construct, for each integral $c \leq O((\log\log n)/(\log\log\log n))$, an EDP instance of size $O(n \log n)$ for which the integrality gap is $\Omega((\frac{\log n}{(\log\log n)^2})^{1/c}/c)$ when congestion is restricted to be *strictly less than* c. Our construction will use two parameters, $\beta_1 = \frac{1}{4}(\frac{\log n}{150(\log\log n)^2})^{1/c}$ and $\beta_2 = 6(2\beta_1)^{c-1}\ln\beta_1$. The integrality gap of our EDP instance will be $\Omega(\beta_1/c)$. Towards the end, we sketch how to extend these results to ANF with congestion.

4.1 Auxiliary Hypergraph Construction

Our starting point is a random hypergraph H with vertex set $V(H) = \{v_1, \ldots, v_n\}$, and $\beta_2 n$ hyper-edges, $h_1, \ldots, h_{n\beta_2}$. Each hyper-edge h_i, for $1 \leq i \leq n\beta_2$ is a c-tuple of vertices, chosen randomly and independently. Our EDP instance will be derived from the hypergraph H.

We now establish some properties of H. Let $S \subseteq V(H)$ be a subset of vertices of size n/β_1. We say that S is *bad* if it contains none of the $n\beta_2$ hyper-edges. We say that event $\mathcal{E}_1$ happens, if there is at least one bad subset $S \subseteq V(H)$ of size n/β_1.

Lemma 3 *The probability that $\mathcal{E}_1$ happens is at most* $1/4$.

Proof: Fix some subset $S \subseteq V(H)$ of size n/β_1 The probability that a random hyper-edge is contained in S is:

$$\frac{\binom{n/\beta_1}{c}}{\binom{n}{c}} = \frac{\frac{n}{\beta_1}\cdot\left(\frac{n}{\beta_1}-1\right)\cdots\left(\frac{n}{\beta_1}-c+1\right)}{n\cdot(n-1)\cdots(n-c+1)} \geq \left(\frac{\frac{n}{\beta_1}-c}{n}\right)^c \geq \frac{1}{(2\beta_1)^c}$$

Therefore,

$$\Pr[S \text{ is bad}] \leq \left(1-\frac{1}{(2\beta_1)^c}\right)^{\beta_2 n} \leq e^{-\frac{\beta_2 n}{(2\beta_1)^c}}$$

Since number of possible sets S is $\binom{n}{n/\beta_1}$ which can be upper-bounded by $(e\beta_1)^{n/\beta_1} \leq \beta_1^{2n/\beta_1}$, using the union bound, we get that the probability that any set S of size n/β_1 is bad, is at most:

$$\beta_1^{\frac{2n}{\beta_1}} \cdot e^{\frac{-\beta_2 n}{(2\beta_1)^c}} \leq e^{\frac{n}{\beta_1}(2\ln\beta_1 - \frac{\beta_2}{2^c\beta_1^{c-1}})} \leq e^{-\frac{n\ln\beta_1}{\beta_1}} \leq \frac{1}{4}$$

□

Given a vertex $v \in V(H)$, we say that it is a *high-degree* vertex, if it participates in more than $10\beta_2 c$ hyper-edges in H. We say that event $\mathcal{E}_2$ happens, if the number of high-degree vertices in H is greater than n/β_1. Using Chernoff bounds, we can show the following.

Lemma 4 *The probability of $\mathcal{E}_2$ happening is at most $1/4$.*

Proof: A vertex v occurs in a random c-tuple with probability c/n. Thus the expected number of hyper-edges in which a vertex is contained is $\beta_2 c$. By Chernoff bounds, for any $\delta \geq 2e-1$, the probability that a vertex is contained in more than $(1+\delta)\beta_2 c$ hyper-edges can be bounded by

$$1/2^{(1+\delta)\beta_2 c} < 1/(4\beta_1).$$

The expected number of high degree vertices is at most $n/(4\beta_1)$. By Markov inequality, the probability that there are more than n/β_1 such vertices is at most $\frac{1}{4}$.

□

4.2 The EDP Instance

The construction of the EDP instance G is based on hyper-graph H defined above. For each vertex $v \in V(H)$, graph G contains a source and sink pair $(s(v), t(v))$. Additionally, for each hyper-edge $h_i : 1 \leq i \leq \beta_2 n$, it contains two vertices ℓ_i, r_i, which are connected by a *special edge*. Consider now some vertex $v \in V$, and assume it participates in hyper-edges $h_{i_1}, h_{i_2}, \ldots, h_{i_k}$, where $i_1 < i_2 < \cdots < i_k$. We add the following *regular edges* to graph G: $(s(v), \ell_{i_1})$, $(r_{i_k}, t(v))$, and for each $j : 1 \leq j \leq k-1$, we add a regular edge $(r_{i_j}, \ell_{i_{j+1}})$. We define a *canonical path* corresponding to v as follows: $P(v) = (s(v), \ell_{i_1}, r_{i_1}, \ldots, \ell_{i_k}, r_{i_k}, t(v))$.

Properties of the EDP Instance: We will establish here that with high probability, the instance created above satisfies some properties that would be useful in establishing our gap.

Let $g > 2$ be some fixed integer, and let K_g be the total number of cycles of length at most g in G. We say that event $\mathcal{E}_3$ happens, if $K_g > (6\beta_2 c^2)^{g+1}$.

Lemma 5 *The probability that $\mathcal{E}_3$ happens is at most $\frac{1}{4}$.*

Proof: Let G' be a graph obtained from G by shrinking each special edge (ℓ_i, r_i) into a vertex u_i, and let K'_g be the number of cycles of length at most g in G'. Since $K_g \leq K'_g$, it is enough to bound K'_g.

Notice that all the source and sink vertices in G have degree 1, and thus do not participate in any cycle. A cycle C of length k in graph G' is defined as an ordered k-tuple of vertices $u_{i_1}, \ldots, u_{i_k}$, where $i_k = \max\{i_1, \ldots, i_k\}$, and edges $e_1 = (u_{i_1}, u_{i_2}), \ldots, e_{k-1} = (u_{i_{k-1}}, u_{i_k}), e_k = (u_{i_k}, u_{i_1})$ belong to G' For each $j : 1 \leq j \leq k-2$, we bound the probability that edge e_j exists given the existence of edges $e_1, \ldots, e_{j-1}$. Let $A \subseteq V(H)$ be the c-tuple of vertices participating in hyper-edge h_{i_j}. If edge e_j exists, then hyper-edge $h_{i_{j+1}}$ must contain at least one vertex from A. The probability of this happening (given the existence of $e_1, \ldots, e_{j-1}$) is at most $\frac{c^2}{n}$.

We now bound the probability of edges e_k, e_{k-1} belonging to G', given the existence of $e_1, \ldots, e_{k-2}$. Consider the hyper-edges $h_{i_1}, h_{i_{k-1}}$ of graph H, and let X, Y, Z be disjoint subsets of $V(H)$, such that $X \cup Y$ are the vertices participating in h_{i_1}, and $Y \cup Z$ are the vertices participating in $h_{i_{k-1}}$. Notice that if hyper-edge h_k contains only vertices belonging to Y (but not to X or Z), then at least one of the edges e_{k-1}, e_k does not belong to G' (this follows from the fact that the canonical path of each vertex $v \in V(H)$ traverses the hyper-edges of H monotonically). Therefore, in order for edges e_{k-1}, e_k to belong to G', hyper-edge h_k must overlap with at least two out of the three sets X, Y, Z. We bound the probability that it overlaps with both X and Y. The probabilities of h_k overlapping with X and Z and with Y and Z are bounded similarly.

Let $\mathcal{X}$, $\mathcal{Y}$ be the events that $h_{i_k} \cap X \neq \emptyset$ and $h_{i_k} \cap Y \neq \emptyset$, respectively. Then:

$$\mathbf{Pr}\left[\mathcal{X} \wedge \mathcal{Y} | e_1, \ldots, e_{k-1}\right] = \mathbf{Pr}\left[\mathcal{X} | \mathcal{Y}, e_1, \ldots, e_{k-1}\right] \cdot \mathbf{Pr}\left[\mathcal{Y} | e_1, \ldots, e_{k-1}\right] \leq \mathbf{Pr}\left[\mathcal{X} | e_1, \ldots, e_{k-1}\right] \cdot \mathbf{Pr}\left[\mathcal{Y} | e_1, \ldots, e_{k-1}\right] \leq \frac{c^4}{n^2}$$

Therefore, the total probability that both edges e_{k-1}, e_k belong to G' is at most $3\frac{c^4}{n^2}$, and the probability that cycle C of length k belongs to G' is at most: $3\left(\frac{c^2}{n}\right)^k$.

The number of potential cycles of length k can be bounded by $(\beta_2 n)^k$. Thus, the expected number cycles of length k is at most $(3\beta_2 c^2)^k$. Summing up over all $k : 3 \leq k \leq g$, we get that $E[K_g] \leq (3\beta_2 c^2)^{g+1}$, and using Markov's inequality, we get the claimed bound.

□

With probability at least 1/4, none of the events $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ happen; we assume this from now on.

4.3 Integrality Gap Analysis

The fractional solution can route at least $\frac{n}{c}$ units of flow, by sending $\frac{1}{c}$ units of flow on each canonical path.

Consider now some integral solution whose congestion is at most $c-1$, and let $\mathcal{P}$ denote the set of paths routed in the integral solution. Set $g = 3\beta_1\beta_2 c^2$. We partition $\mathcal{P}$ into three subsets: $\mathcal{P}_1$ contains canonical paths, $\mathcal{P}_2$ contains non-canonical paths whose length is greater than g, and $\mathcal{P}_3$ contains non-canonical paths whose length is smaller than g. We bound the size of each one of these sets separately.

$|\mathcal{P}_1| \leq \frac{n}{\beta_1}$ if event $\mathcal{E}_1$ does not happen. Otherwise, there must be c paths that go through a single special edge and the solution has congestion c.

$|\mathcal{P}_2| \leq n/\beta_1$. Total number of edges in G can be bounded by $3\beta_2 cn$. Since we allow a congestion of c, total capacity available in the graphs is at most $3\beta_2 c^2 n$. Thus the number of paths of length greater than g can be no more than $\frac{3\beta_2 c^2 n}{g} = \frac{3\beta_2 c^2 n}{3\beta_1\beta_2 c^2} = \frac{n}{\beta_1}$.

To analyze $\mathcal{P}_3$, we first remove from $\mathcal{P}_3$ all paths that correspond to vertices which occur in more than $10\beta_2 c$ hyper-edges of H. Since event $\mathcal{E}_2$ does not happen, we discard at most n/β_1 paths. Let $\mathcal{P}'_3$ be the set that remains. For any $s(v)$-$t(v)$ pair routed in $\mathcal{P}'_3$, the length of its canonical path is at most $10\beta_2 c$. Thus the non-canonical path in $\mathcal{P}'_3$ and the canonical path for $s(v)$-$t(v)$ form a cycle of length at most $g + 10\beta_2 c \leq 2g$. By Lemma 5, the number of cycles of length at most $2g$ can be bounded by $(6\beta_2 c)^{2g+2}$. Since each edge is allowed a congestion of up to $(c-1)$, and each path in $\mathcal{P}'_3$ uses an edge on such a cycle, $|\mathcal{P}'_3|$ is bounded by $2gc(6\beta_2 c)^{2g+2}$.

Therefore, $|\mathcal{P}'_3|$ can be bounded as follows:

$$\begin{aligned}
|\mathcal{P}'_3| &\leq 2gc(6\beta_2 c^2)^{2g+2} \\
&\leq (\beta_2 c^2)^{3g} \\
&\leq 2^{4g \log \beta_2} \\
&= 2^{12\beta_1\beta_2 c^2 \log \beta_2} \\
&\leq 2^{12\beta_1 \cdot 6(2\beta_1)^{c-1} \ln \beta_1 \cdot 2c \ln \beta_1} \\
&= 2^{72(4\beta_1)^c \cdot \ln^2 \beta_1} \\
&\leq \sqrt{n} \\
&\leq n/\beta_1
\end{aligned}$$

In total, $|\mathcal{P}| \leq 4n/\beta_1$, and the integrality gap is at least $\frac{\beta_1}{4c}$, giving the bound in Theorem 2.

To show the hardness of ANF with congestion, classify each routed pair to be of type A or B based on how much flow is routed on canonical versus non-canonical paths. It is type A if more than a $(c-1)/c$-fraction of the flow is routed on the pair's canonical path, and type B otherwise. It is easy to see that no more than $(c-1)$ type A pairs can traverse a special edge without causing a congestion greater than $c-1$. Thus essentially the same analysis as given above for $\mathcal{P}_1$ applies. For type B pairs, we proceed as above for $\mathcal{P}_2$ and $\mathcal{P}_3$ noting that for each routed pair, we have only $1/c$-fraction of the flow to be supported.

5 Hardness of EDP with Congestion

We will now establish hardness of approximating EDP with congestion $c \geq 2$. We will focus here on the case when c is any constant. As earlier, we perform a reduction from 3SAT using the PCP characterization presented in Section 2. The parameters q, λ, and r stay the same when c is a constant. Towards the end, we briefly describe how the parameters change when c is allowed to be as large as $O(\log\log n)/(\log\log\log n)^2$.

In what follows, let z be the least integer such that $c < 2^z$. We will iteratively define sample spaces of EDP instances, namely $H_1, H_2, ..., H_z$, such that the sample space H_i is defined in terms of H_{i-1} for $2 \leq i \leq z$. The starting sample space H_1 is identical to one described in Section 3. We will prove that if ϕ is a YES-INSTANCE , then any instance of H_z has a collection of edge disjoint paths in H_z of size at least P_{YI}, while if ϕ is a No-Instance, then with high probability, at most P_{NI} source-sink pairs can be routed with congestion restricted to $2^z - 1$. We show that for any constant $\epsilon : 0 < \epsilon < 1$, $P_{YI}/P_{NI} \geq (\log N)^{(1-\epsilon)/(\frac{3}{2}2^z - 1)}$, where N is the size of instance H_z.

5.1 Construction

We will use as our building block the bit gadget in Section 3.1. We will vary the parameters M, X based on the sample space H_i. The sample space H_1 is same as in Section 3, except that instead of parameters M, X, we use new parameters M_1, X_1, which are specified later. For each accepting configuration α, let $\mathcal{P}^1_\alpha$ denote the set of X_1 canonical paths representing α in H_1. For $i \geq 2$, we generate an instance of H_i by connecting together several random instances of H_{i-1}. Graph H_i will contain a set of regular edges and a set of special edges, whose sizes are the same for all the instances of H_i. An instance of H_i contains X_1 source-sink pairs for each ordered i-tuple $(\alpha_1, \ldots, \alpha_i)$ of accepting configurations. For each pair, there is a canonical path, and let η_i denote the number of canonical paths in any instance of H_i. Clearly, $\eta_i = |\mathcal{C}|^i X_1$. In order to define the recursive construction of H_i, we need first to define the notion of concatenation of instances of H_i.

Concatenation of EDP Instances: Suppose G_1, G_2 are two instances of H_{i-1}, for some $i \geq 2$. Then *concatentation* of G_1 and G_2 is a new instance G defined as follows. Let $(\alpha_1, \ldots, \alpha_{i-1})$ be an ordered $(i-1)$-tuple of accepting configurations. Recall that each instance of H_{i-1} contains X_1 source-sink pairs representing $(\alpha_1, \ldots, \alpha_{i-1})$. Let S_1, T_1 and S_2, T_2 be the corresponding sets of source and sink vertices in G_1 and G_2, respectively. We randomly unify the vertices in T_1, S_2 in a pairwise manner. Consider any two source-sink pairs (s_1, t_1) and (s_2, t_2) corresponding to $(\alpha_1, \ldots, \alpha_{i-1})$ in G_1 and G_2, respectively, such that t_1 and s_2 are unified in G. Then (s_1, t_2) becomes a source-sink pair for graph G, and its canonical path is defined as a concatenation of the two canonical paths in G_1 and G_2. Observe that in graph G, the number of source-sink pairs remains $|\mathcal{C}|^{i-1} X_1$, the same as in G_1 and G_2. We define a concatenation of arbitrary number of instances of H_{i-1} in a similar fashion.

Definition of H_i: An instance of H_i is constructed by a recursive composition of instances of H_{i-1} and bit gadgets. We will use parameter M_i, X_i for constructing H_i. For $i \geq 2$, we define $M_i = M_1^3 M_2 ... M_{i-1}$. Similarly, we define $X_i = (|\mathcal{C}| q M_{i-1} X_{i-1})/2$ for $i \geq 2$. By our choice of parameters, we ensure that the number of special edges in an instance of H_{i-1} is X_i.

- For each accepting configuration α, and for each $j : 1 \leq j \leq q$, we build an instance $B^{i-1}(\alpha, j)$ of H_{i-1}. Each of these instances is constructed independently.
- For each accepting configuration α, we define a graph $G^i(\alpha)$ to be the concatenation of $B^{i-1}(\alpha, 1), \ldots, B^{i-1}(\alpha, q)$. A source-sink pair in the concatenated graph correponding to an $(i-1)$-tuple $(\alpha_1, \ldots, \alpha_{i-1})$ can now be viewed as a pair that corresponds to the i-tuple $(\alpha_1, \ldots, \alpha_{i-1}, \alpha)$ in $G^i(\alpha)$.
- For each proof bit Π_j, we build a bit gadget $G^i(j)$ representing it, with parameters M_i, X_i.
- The above two parts are composed together as follows. Consider some accepting configuration α, and let $a_1, \ldots, a_q$ be the corresponding query bits. Fix some $j : 1 \leq j \leq q$.

  On one hand, we have a bit gadget $G^i(a_j)$, which contains X_i canonical paths corresponding to α. Let S_j, T_j denote the set of sources and destinations of these paths. For each source $s \in S_j$, let $f(s) \in T_j$ denote its corresponding destination.

  On the other hand, graph $G^i(\alpha)$ contains as sub-graph instance $B^{i-1}(\alpha, j)$ of H_{i-1}, which has X_i special edges. Let A denote this set of special edges, and let L and R denote the sets of their left and right endpoints. We remove these edges from our graph. Instead, we unify vertices in L and S_j (in pairwise manner), and we unify vertices in R and T_j, as follows. Let $e = (\ell, r) \in A$, and assume we unified ℓ with some source $s \in S_j$. We then unify r and $f(s)$.

- Source-sink pairs are the union of the source-sink pairs in graphs G^i_α for $\alpha \in \mathcal{C}$.

The set of special edges in the new instance of H_i is the union of the special edges in bit gadgets $G^i(j)$, for all proof bits j. All the other edges are regular. Notice that the number of special edges in H_i is indeed X_{i+1}: Recall that for each configuration $\alpha \in \mathcal{C}$, graph $G^i(\alpha)$ is a concatenation of q instances of H^{i-1}, each of them containing X_i special edges. Each such special edge is replaced by a canonical path in $G^i(j)$ for some proof bit j. A canonical path of a bit gadget has M_i special edges, and each special edge is shared by two such paths. Therefore, the total number of special edges in H_i is $|\mathcal{C}|qX_i\frac{M_i}{2} = X_{i+1}$.

Also, note that the total number of canonical paths that go through a special edge in any instance of H_i is exactly 2^i.

Size of an instance of H_z: We will set the base parameters as $M_1 = 2^{\lambda k^2}$ and $X_1 = 2^{2^{\lambda k^2(\frac{3}{2}2^z-1)+\lambda k}}$. Let us now bound M_i. It is easy to see that $M_2 = M_1^3$, and for all $i : 2 < i \le z$, $M_i = M_{i-1}^2 = M_1^{\frac{3}{4}2^i}$. Therefore, for all $2 < i \le z$, we have $M_i = 2^{\lambda k^2 \frac{3}{4}2^i} < 2^{\lambda k^2 2^i}$.

Let N_i denote the size of an instance of H_i, and let ℓ_i denote the length of each canonical paths in an instance of H_i. Recall that $\eta_i = |\mathcal{C}|^i X_1$ is the number of canonical paths in H_i. Clearly, $N_i \le \ell_i \eta_i$.

To bound ℓ_1, recall that each canonical path traverses q gadgets, and length of a canonical path inside each gadget is at most $3M_1$. So, $\ell_1 \le 4qM_1$. The recursive formula for ℓ_i, where $i > 1$ is calculated as follows. A canonical path in H_i consists of q canonical paths in H_{i-1}. Additionally, each special edge of H_{i-1} is replaced with a canonical path in gadget $G^i(j)$ (where j is some proof bit index). The length of the canonical path inside $G^i(j)$ is at most $3M_i$, and the number of special edges on path ℓ_i is at most $q\ell_{i-1}$. Therefore, $\ell_i \le q\ell_{i-1} + q\ell_{i-1}\cdot 3M_i \le 4qM_i\ell_{i-1} \le (4q)^i M_1 M_2 \cdots M_i \le (4q)^i M_1^{\frac{3}{4}2^{i+1}-2}$. Thus the size N_z of an instance of H_z can be bounded as $N_z \le \ell_z \eta_z \le |\mathcal{C}|^z X_1 (4q)^z M_1^{\frac{3}{4}2^{z+1}-2} \le X_1 2^{2\lambda r z}$ when $z = O(\log\log\log n)$.

Notice that $X_1 = 2^{2^{\lambda k^2(\frac{3}{2}2^z-1)+\lambda k}}$. As $r = O(\log n)$, the overall construction size is $O(n^{\operatorname{poly}\log n})$.

5.2 Yes Instance: ϕ is Satisfiable

In the Yes-Instance , there is a PCP proof, for which the acceptance probability is at least $2^{-\lambda}$. For each random string r satisfied by this proof, let $c(r)$ be the corresponding accepting configuration.

Lemma 6 *If ϕ is a* Yes-Instance *, then for each $i : 1 \le i \le z$, graph H_i contains a collection of $P_{YI} = |\mathcal{C}|^i X_1 / 2^{(2\lambda k+\lambda)i}$ edge-disjoint canonical paths.*

Proof: Let $\mathcal{P}^i_{YI}$ be the collection of canonical paths, defined as follows. Consider any i-tuple $(\alpha_1, \ldots, \alpha_i)$ of accepting configurations. The set $\mathcal{P}^i_{YI}$ contains the X_1 paths representing this i-tuple in H_i, iff for each $j : 1 \le j \le z$, $\alpha_j = c(r_j)$ for some random string r_j. Therefore, $|\mathcal{P}^i_{YI}| \ge (|R|^i X_1)/2^{\lambda i} \ge (|\mathcal{C}|^i X_1)/2^{(2\lambda k+\lambda)i}$, where the last inequality follows from the property that for each random string r, there are at most $2^{2\lambda k}$ accepting configurations. We now prove, by induction on i, that for all $i : 1 \le i \le z$, all the paths in $\mathcal{P}^i_{YI}$ are edge-disjoint in H_i.

For $i = 1$, since there is no conflict between any pair of configurations $c(r)$, $c(r')$ for random strings r, r', all the paths in $\mathcal{P}^1_{YI}$ are edge-disjoint. Assume now that all the paths in $\mathcal{P}^{i-1}_{YI}$ are edge-disjoint in H_{i-1}, and consider H_i. Let α be any accepting configuration. If $\alpha \ne c(r)$, where r is the random string of α, then none of the paths in G^i_α belongs to our solution. Therefore, if we have two accepting configurations α, β, such that some paths in G_α and some paths in G_β belong to the solution, then there is no conflict between α and β, and thus they will not participate together in the same gadget $G^i(j)$ for any proof bit j. Therefore, the only way for the solution not to be edge-disjoint is that for some configuration α, there are two paths of G_α in our solution, which share an edge. But G_α is a concatenation of q instances of H_{i-1}, and when restricted to each one of these instances, our solution is exactly the set $\mathcal{P}^{i-1}_{YI}$ of paths, which is edge-disjoint for any instance of H_{i-1} by the induction hypothesis. □

5.3 No Instances: ϕ is Unsatisfiable

Assume ϕ is a NO-INSTANCE . As before, we will bound the number of canonical paths ($\mathcal{P}_1$), long non-canonical paths ($\mathcal{P}_2$), and short canonical paths ($\mathcal{P}_3$) in any solution that has congestion at most $2^z - 1$.

5.3.1 Canonical Paths

Recall that in order to construct our final graph H_z, we construct, for each proof bit Π_j, for each $i : 1 \le i \le z$, many instances of bit gadget $G^i(j)$, with parameter M_i. We define a parameter $\Delta_i = \frac{M_i}{8 \log M_i}$, which replaces the parameter Δ in the definition of a bad gadget. Let $\mathcal{B}_1$ be the (bad) event that any of these bit gadgets is bad. The following is a simple corollary of Lemma 1.

Corollary 2 *The probability of the bad event $\mathcal{B}_1$ is bounded by $\frac{1}{\text{poly}\, n}$.*

Proof: From Lemma 1, the probability that any gadget is bad is at most e^{-n}. Since our construction size is quasi-polynomial, it contains less than $e^{n/2}$ bit gadgets. The corollary follows by the union bound. □

Theorem 4 *If event $\mathcal{B}_1$ does not occur, then for each $i : 1 \le i \le z$, any collection of more than $\frac{|\mathcal{C}|^i \cdot (9q^2)^i \cdot X_1}{M_1}$ canonical paths in graph H_i, causes congestion of 2^i.*

Proof: The proof is by induction on i. For each i, we bound the maximum number of canonical paths, for which congestion is less than 2^i. The analysis of the base case, where $i = 1$ is similar to the analysis presented in Section 3. Recall that the number of canonical paths in any solution with congestion 1 is at most $\frac{X_1}{\Delta_1} \sum_j n_j + \frac{2^{\lambda r} X_1}{2^{\lambda k^2}} \le \frac{|\mathcal{C}| q X_1 \cdot 8 \log M_1}{M_1} + \frac{|\mathcal{C}| X_1}{2^{\lambda k^2}} \le \frac{|\mathcal{C}| q X_1 \cdot 8 \lambda k^2}{M_1} + \frac{|\mathcal{C}| X_1}{2^{\lambda k^2}} \le \frac{|\mathcal{C}| (9q^2) X_1}{M_1}$.

Assume now the theorem holds for $i - 1$, and consider H_i. Let $\mathcal{P}_1^i$ be any collection of canonical paths in H_i, such that their congestion is less than 2^i. We partition the set $\mathcal{P}_1^i$ as follows: for each configuration $\alpha \in \mathcal{C}$, let $\mathcal{Q}_\alpha^i$ be the paths of $\mathcal{P}_1^i$ that correspond to paths in $G^i(\alpha)$.

<u>Definition</u>: Let $\alpha \in \mathcal{C}$ be an accepting configuration. We say that α is *congested* iff $|\mathcal{Q}_\alpha^i| \ge 2\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1}$.

We now proceed in two steps. First, we prove that if α is congested, then for each $j : 1 \le j \le q$, many of the special edges in $B^{i-1}(\alpha, j)$ have congestion 2^{i-1}. The second step is proving that the number of congested configurations is small (otherwise the overall congestion is 2^i).

Lemma 7 *Suppose α is congested and event $\mathcal{B}_1$ does not occur. Then for each $j \in \{1..q\}$, at least $\frac{X_i}{M_1^2 M_2 \cdots M_{i-1}}$ special edges in instance $B^{i-1}(\alpha, j)$ of H_{i-1}, have congestion 2^{i-1}.*

Proof: Fix j and consider instance $B^{i-1}(\alpha, j)$ of H_{i-1}. Let $\mathcal{Q}^i(\alpha, j)$ denote the restriction of $\mathcal{Q}_\alpha^i$ to the canonical paths of $B^{i-1}(\alpha, j)$. We refer to the special edges of $B^{i-1}(\alpha, j)$ whose congestion is 2^{i-1} as *congested edges*.

Since α is a congested configuration, $|\mathcal{Q}_\alpha^i| \ge 2\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1}$. However, by the induction hypothesis, the maximum number of canonical paths in any instance of H_{i-1} that do not cause congestion 2^{i-1} is $\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1}$. In order to convert $\mathcal{Q}^i(\alpha, j)$ into a collection of canonical paths that cause congestion less than 2^{i-1}, we need to remove at least $\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1}$ paths from it. Each such removed path reduces congestion on at least one congested edge. Therefore, the number of congested special edges is at least $\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1}$. Since the total number of special edges in any instance of H_{i-1} is $X_i \le N_{i-1} \le |\mathcal{C}|^{i-1} X_1 (9q^2)^{i-1} M_1 \cdots M_{i-1}$, the number of congested edges is at least $\frac{X_i}{M_1^2 M_2 \cdots M_{i-1}}$. □

Lemma 8 *If ϕ is a* NO-INSTANCE *and the event $\mathcal{B}_1$ does not occur, then in any solution of H_i with congestion at most $2^i - 1$, no more than $\frac{2^{i+1} |\mathcal{C}| q^2}{M_1}$ configurations can be congested.*

Proof: Let $\mathcal{C}_1 \subseteq \mathcal{C}$ denote the subset of congested configurations. Consider some bit gadget $G^i(j)$. Recall that each canonical path in gadget $G^i(j)$ corresponds to some special edge of some $G^i(\alpha)$ for $\alpha \in \mathcal{C}$. Let $e_1 \in G^i(\alpha), e_2 \in G^i(\gamma)$ be two special edges, and assume that in the construction of H_i, edge e_1 was replaced by the canonical path $P_x(\alpha)$, and e_2 was replaced by the canonical path $P_{x'}(\gamma)$ in gadget $G^i(j)$. Then if these canonical paths share an edge in $G^i(j)$, and if edges e_1 and e_2 are congested, our solution has total congestion 2^i.

Since gadget $G^i(j)$ is not bad, and congestion is less than 2^i, either there are less than $\frac{n_j X_i}{\Delta_i}$ congested special edges that belong to graphs $G^i(\alpha)$ where $\alpha \in Z_j$, or there are less than $\frac{n_j X_i}{\Delta_i}$ congested special edges in graphs $G^i(\alpha)$ where $\alpha \in O_j$ (here $\Delta_i = M_i/8\log M_i$). Recall that for each congested configuration, there are at least $\frac{X_i}{M_1^2 M_2 \cdots M_{i-1}}$ congested special edges. Therefore, either Z_j contains less than $\frac{n_j M_1^2 M_2 \cdots M_{i-1}}{\Delta_i}$ congested configurations, or O_j contains less than $\frac{n_j M_1^2 M_2 \cdots M_{i-1}}{\Delta_i}$ congested configurations. Thus by removing at most $\sum_j n_j \frac{M_1^2 M_2 \cdots M_{i-1}}{\Delta_i}$ configurations from $\mathcal{C}_1$, we obtain a subset $\mathcal{C}_2$ of accepting configurations with no conflicts. As we have a NO-INSTANCE , $|\mathcal{C}_2| \leq \frac{|\mathcal{C}|}{2^{\lambda k^2}}$.

On the other hand, we can bound $\sum_j n_j \leq |\mathcal{C}|q$, and thus,

$$|\mathcal{C}_1 \setminus \mathcal{C}_2| \leq \frac{|\mathcal{C}|q M_1^2 M_2 \cdots M_{i-1}}{\Delta_i} = \frac{|\mathcal{C}|q M_1^2 M_2 \cdots M_{i-1} \cdot 8 \log M_i}{M_i} \leq \frac{|\mathcal{C}|q\lambda k^2 2^i}{M_1} \leq \frac{|\mathcal{C}|q^2 2^i}{M_1}$$

(we have used the fact that $M_i = M_1^3 M_2 \cdots M_{i-1}$, and that $\log M_i < \lambda k^2 2^i$).

In total, $|\mathcal{C}_1| \leq \frac{|\mathcal{C}|q^2 2^i}{M_1} + \frac{|\mathcal{C}|}{2^{\lambda k^2}} \leq \frac{2^{i+1}|\mathcal{C}|q^2}{M_1}$ □

We are now ready to bound the number of canonical paths in $\mathcal{P}_1 = \mathcal{P}_1^i$. Each congested configuration contributes at most $|\mathcal{C}|^{i-1} X_1$ paths to $\mathcal{P}_1$, and by Lemma 8, we have at most $(2^{i+1}|\mathcal{C}|q^2)/M_1$ congested configurations. Each non-congested configuration contributes at most $2\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1}$ paths. Thus, $|\mathcal{P}_1^i| \leq |\mathcal{C}|^{i-1} X_1 \cdot \frac{2^{i+1}|\mathcal{C}|q^2}{M_1} + 2\frac{|\mathcal{C}|^{i-1} \cdot (9q^2)^{i-1} \cdot X_1}{M_1} \cdot |\mathcal{C}| \leq \frac{2^{i+1}|\mathcal{C}|^i q^2 X_1}{M_1} + 2\frac{|\mathcal{C}|^i \cdot (9q^2)^{i-1} \cdot X_1}{M_1} \leq \frac{|\mathcal{C}|^i X_1 (9q^2)^i}{M_1}$. □

Since $\mathcal{P}_1 = \mathcal{P}_1^z$, we get $\frac{P_{YI}}{|\mathcal{P}_1|} \geq \frac{2^{\lambda k^2}}{(9q^2)^z \cdot 2^{(2\lambda k + \lambda)z}} \geq 2^{\lambda k^2 - 2\lambda kz - \lambda z - 3z \log(\lambda k)} \geq 2^{\lambda k^2 - 3\lambda kz}$.

5.3.2 Long Non-Canonical Paths

Recall that the length of each canonical path in an instance of H_z is $\ell_z \leq (4q)^z M_1^{\frac{3}{4}(2^{z+1}-2)} \leq (4q)^z 2^{\lambda k^2 \frac{3}{4}(2^{z+1}-2)}$. A non-canonical path is called *long* if its length is at least $g = \ell_z \gamma$ where $\gamma = 2^{\lambda k^2}$. Otherwise, it is called *short*. Let $\mathcal{P}_2$ denote the set of long non-canonical paths in any solution that has congestion less than c. Each edge in our final graph participates in at least one canonical path. Thus, the total number of edges is at most $\eta_z \ell_z$. As the congestion on each edge is less than 2^z, we have that $|\mathcal{P}_2| \leq \frac{2^z \eta_z \ell_z}{g} = \frac{2^z |\mathcal{C}|^z X_1}{\gamma} \leq P_{YI} \frac{2^z 2^{(2\lambda k+\lambda)z}}{2^{\lambda k^2}} \leq \frac{P_{YI}}{2^{\lambda(k^2-3kz)}}$.

5.3.3 Short Non-Canonical Paths

We next bound the size of $\mathcal{P}_3$, the set of short non-canonical paths in our solution. Suppose there is a short non-canonical path $P \in \mathcal{P}_3$ connecting some source and destination pair (s,t). This path must form a cycle of length at most $g + \ell_z \leq 2g$ with the canonical $s-t$ path. Moreover, at least one edge on the cycle lies on P. Let K denote the number of cycles of length at most $2g$ in our graph. Then $|\mathcal{P}_3| \leq 2cg \cdot K$ (since congestion is at most c). Our goal is to show that with high probability, K is small. The proof of the following lemma is similar to Lemma 2.

Lemma 9 *With probability at least $\frac{2}{3}$, $K \leq 3|\mathcal{C}|^{(2g+2)z}$.*

Proof: We build a new graph G', obtained by shrinking the special edges (each special edge becomes a vertex). Let K' denote the number of cycles of length at most $2g$ in G'. Clearly, $K \leq K'$. We now bound K'.

All the potential edges whose probability is greater than 0 in our graph can be partitioned into several disjoint subsets, where in each subset we perform a matching between two sets of X_i vertices for $1 \leq i \leq z$. The different matchings are completely independent. Therefore, the probability that some edge e exists in graph G' is at most $\frac{1}{X_1}$ (since $X_1 \leq X_i$ for all $i \geq 1$), and the probability that e exists given the existence of i other edges, $1 \leq i \leq 2g$ is at most $\frac{1}{X_1 - 2g} \leq \frac{2}{X_1}$. Thus, the probability that a given potential cycle that contains g' edges, $g' : 1 \leq g' \leq 2g$ exists is at most $\left(\frac{2}{X_1}\right)^{g'}$. The number of potential cycles of length g' is bounded by $N_z^{g'}$, where N_z is the graph size. Thus the expected number of cycles of length g' in G' is at most $\left(\frac{2N_z}{X_1}\right)^{g'}$. Summing up over all values of g', the expected number of cycles of length at most $2g$ is at most $\left(\frac{2N_z}{X_1}\right)^{2g+1}$. Since N_z is bounded by $\eta_z \ell_z$, we get

$$E(K') \leq \left(\frac{2\eta_z \ell_z}{X_1}\right)^{(2g+2)} \leq |\mathcal{C}|^{(2g+2)z}$$

Using Markov's inequality, the probability that the number of cycles of length at most $2g$ is greater than $3|\mathcal{C}|^{(2g+1)z}$ is at most 1/3. □

We say that the *bad event* $\mathcal{B}_2$ occurs if $K > 3|\mathcal{C}|^{(2g+2)z}$. If $\mathcal{B}_2$ does not happen, then we have: $|\mathcal{P}_3| \leq 2cg \cdot (3|\mathcal{C}|^{(2g+2)z}) \leq |\mathcal{C}|^{3gz} \leq 2^{(\lambda(r+2k)z)\cdot(4q)^z 2^{\lambda k^2(\frac{3}{4}2^{z+1}-2)} \cdot 2^{\lambda k^2}} \leq 2^{2^{\lambda k^2(\frac{3}{4}2^{z+1}-1)+\lambda k}} \leq X_1 \leq \frac{P_{YI}}{2^{\lambda(k^2-3kz)}}$.

5.3.4 Putting it Together

If the events $\mathcal{B}_1$ and $\mathcal{B}_2$ do not happen, the gap between the yes and the no instances is $\Omega(2^{\lambda(k^2-3kz)})$. For any $\epsilon' > 0$, we can choose sufficiently large k such that the gap is $(\log N_z)^{(1-\epsilon')/(\frac{3}{4}2^{z+1}-1)} = (\log N_z)^{(1-\epsilon')/(\frac{3}{2}c+\frac{1}{2})}$ and the size of the instance is bounded by $n^{\text{polylog(n)}}$.

When c is allowed to grow up to $\frac{\log\log n}{(\log\log\log n)^2}$ for any $\epsilon > 0$, the gap term $\Omega(2^{\lambda(k^2-3kz)})$ yields the desired gap only when we allow k to grow to $z = \log\log\log n$. Following [20], it can be shown that using $r = O(k^2 \log n)$ random bits, we can get once again completeness at least 1/2 and soundness at most $1/2^{k^2}$. To keep the construction size bounded by $(n^{\text{polylog(n)}})$, we now choose λ to be a large constant. The rest of the proof remains similar to the one presented above.

Acknowledgements

We thank Madhu Sudan for many useful discussions.

References

[1] M. Andrews and L. Zhang. Hardness of the Undirected Edge-Disjoint Paths Problem. *Proc. of STOC*, 2005.

[2] M. Andrews and L. Zhang. Hardness of the Edge-Disjoint Paths with Congestion. http://cm.bell-labs.com/˜andrews/pub.html, 2005.

[3] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, February 1998.

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

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

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

[7] C. Chekuri, S. Khanna, and F. B. Shepherd. Edge Disjoint Paths in Planar Graphs. *Proc. of FOCS*, 2004.

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

[9] C. Chekuri, S. Khanna, and F. B. Shepherd. An $O(\sqrt{n})$-approximation for EDP in Undirected Graphs and Directed Acyclic Graphs. *Manuscript*, 2005.

[10] C. Chekuri and S. Khanna. On Multidimensional Packing Problems. To appear in *SIAM J. on Computing*. Preliminary version in *Proc. of SODA*, 1999.

[11] C. Chekuri and S. Khanna. Edge Disjoint Paths Revisited. *Proc. of SODA*, 2003.

[12] C. Chekuri, M. Mydlarz, and F. B. Shepherd. Multicommodity Demand Flow in a Tree and Packing Integer Programs. Submitted. Preliminary version in *Proc. of ICALP*, 2003.

[13] S. Even, A. Itai and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. on Computing*, Vol 5, 691-703, 1976.

[14] A. Frank. Edge-disjoint paths in planar graphs. *J. of Combinatorial Theory, Ser. B.*, No. 2 (1985), 164-178.

[15] S. Fortune, J. Hopcroft and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, Vol. 10, No. 2 (1980), pp. 111–121.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.

[17] N. Garg, V. Vazirani, M. Yannakakis. Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees. *Algorithmica*, 18(1):3-20, 1997. Preliminary version appeared in *Proc. of ICALP*, 1993.

[18] 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. To appear in *JCSS*. Preliminary version appeared in *Proc. of STOC*, 1999.

[19] V. Guruswami and K. Talwar. *In preparation.*

[20] J. Hastad, A. Wigderson. Simple Analysis of Graph Tests for Linearity and PCP. Random Structures and Algorithms, Vol 22, no. 2, pp 139-160, 2003.

[21] R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, Eds., New York: Plenum Press, 1972, 85–103.

[22] J. M. Kleinberg and É. Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. *Journal of Computer and System Sciences*, 57:61–73, 1998. Preliminary version in the *Proc. of STOC*, 1995.

[23] J. M. Kleinberg and É. Tardos. Disjoint Paths in Densely Embedded Graphs. *Proc. of FOCS*, pp. 52–61, 1995.

[24] J. M. Kleinberg. Approximation algorithms for disjoint paths problems. PhD thesis, MIT, Cambridge, MA, May 1996.

[25] J. M. Kleinberg. Decision algorithms for unsplittable flow and the half-disjoint paths problem. *Proc. of STOC*, pp. 530–539, 1998.

[26] S. G. Kolliopoulos and C. Stein. Improved approximation algorithms for unsplittable flow problems. *Proc. of FOCS*, 426–435, 1997.

[27] S. G. Kolliopoulos and C. Stein. Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs. IPCO 1998: 153-168.

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

[29] N. Robertson and P. D. Seymour. Outline of a disjoint paths algorithm. In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, Eds., *Paths, Flows and VLSI-Layout.* Springer-Verlag, Berlin, 1990.

[30] A. Samorodnitsky and L. Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the 32nd Annual ACM Symposium on theory of Computing*, 2000.

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

[32] K. Varadarajan and G. Venkataraman. Graph Decomposition and a Greedy Algorithm for Edge-disjoint Paths. *Proc. of SODA*, 2004.

A The Samorodnitsky-Trevisan PCP Construction

Our starting point is the following PCP characterization of NP, proved by Samorodnitsky and Trevisan in [30]. Let ϕ be an instance of 3SAT on n variables. We have a PCP proof for the satisfiability of ϕ denoted by Π and a randomized verifier, that reads q proof bits to determine whether or not ϕ is satisfiable. Given a random string r of the verifier, let $b_1(r),\ldots,b_q(r)$ be the indices of the proof bits read. A *configuration* is $(r,a_1,\ldots,a_q)$, where $a_1,\ldots,a_q \in \{0,1\}$ are values of $\Pi_{b_1(r)},\ldots,\Pi_{b_q(r)}$. We say that a configuration $(r,a_1,\ldots,a_q)$ is *accepting*, if, for a random string r of the verifier and the values $a_1,\ldots,a_q$ of proof bits $\Pi_{b_1(r)},\ldots,\Pi_{b_q(r)}$, the verifier accepts. Abusing the notation, we denote by r both the random string of the verifier and the number of random bits (i.e., the length of the string).

Theorem 5 *[30] For any constants $\mu > 0, k > 0$, there is a PCP verifier for 3SAT with the following properties:*

- *If ϕ is a* Yes-Instance *, accepts with probability $\geq 1-\mu$.*
- *If ϕ is a* No-Instance *, accepts with probability $\leq \mu + 2^{-k^2}$.*
- *Reads $2k+k^2$ query bits and tosses $r = O(\log n)$ random coins.*
- *For every random string r, there are 2^{2k} accepting configurations.*
- *For every random string r, for every $i : 1 \leq i \leq q$, the number of accepting configurations where the value of $\Pi_{b_i(r)} = 0$ equals the number of accepting configurations where $\Pi_{b_i(r)} = 1$.*

In our case, we fix $\mu = 2^{-k^2}$ and k is a large enough constant. Thus, in Yes-Instance , the acceptance probability is at least $\frac{1}{2}$ and in No-Instance , it is at most $2 \cdot 2^{-k^2}$.

Observe that the length of the proof is bounded by $2^r(2k+k^2)$. For the sake of convenience, we would like to ensure that each proof bits participates in many accepting configurations. This can be done as follows. The verifier works as before, except that now additionally it randomly chooses one proof bit and accepts iff the original verifier accepts. Observe that now the length of the random string becomes $r' \leq r + r + 3\log k$, and the number of query bits read is k^2+2k+1. Let Π_j be some proof bit, and let n_j be the number of accepting configurations in which the value of Π_j is 0 (clearly, the number of accepting configurations in which the value of Π_j is 1 is also n_j). We have that $n_j \geq 2^r \cdot 2^{2k} \geq 2^{r'/2}$.

We summarize the properties of the PCP construction (substituting the values of r and k by the new values r' and $k' = k+1$):

- The verifier reads at most k^2 query bits and tosses $r = O(\log n)$ random bits.
- YES-INSTANCE : acceptance probability at least $\frac{1}{2}$.
- NO-INSTANCE : acceptance probability at most 2^{-k^2}.
- For every random string, there are 2^{2k-1} accepting configurations.
- For every random string r, for every $j : 1 \leq j \leq q$, the number of accepting configurations where the value of $\Pi_{b_j(r)} = 0$ equals the number of accepting configurations where $\Pi_{b_j(r)} = 1$.
- For each proof bit Π_j, $n_j \geq 2^{r/2}$.

Repetitions

As a next step, we perform $\lambda = O(\log \log n)$ independent repetitions of the above protocol. Recall that k is a large enough constant. Let β be a large constant, $\beta >> k^2$. We fix $\lambda = \frac{2\beta \log \log n}{k^2} = O(\log \log n)$.

The verifier accepts iff the original verifier accepts in each protocol repetition. The resulting PCP has the following properties:

- **Random Bits:** $\lambda r = O(\log n \log \log n)$. Let R denote the set of all possible random string, $|R| = 2^{\lambda r}$.
- **Query Bits:** $q = \lambda k^2 = O(\log \log n)$. W.l.o.g. assume that the verifier reads exactly q bits of proof for every random string.
- **Completeness:** YES-INSTANCE is accepted with probability at least $2^{-\lambda}$
- **Soundness:** NO-INSTANCE is accepted with probability at most $2^{-\lambda k^2}$
- For each random string, there are $2^{\lambda(2k-1)}$ accepting configurations.
- For every random string r, for every $j : 1 \leq j \leq q$, the number of accepting configurations where the value of $\Pi_{b_j(r)} = 0$ equals the number of accepting configurations where $\Pi_{b_j(r)} = 1$.
- For each proof bit Π_j let Z_j be the set of all the accepting configurations in which bit Π_j participates with value 0, and let O_j be the set of all the accepting configurations in which Π_j participates with value 1. We denote $n_j = |Z_j| = |O_j|$. Then $n_j \geq 2^{\lambda r/2}$.