SIAM J. COMPUT.
Vol. 43, No. 2, pp. 497–513

EXPANDERS VIA RANDOM SPANNING TREES*

ALAN FRIEZE†, NAVIN GOYAL‡, LUIS RADEMACHER§, AND SANTOSH VEMPALA¶

Abstract. Motivated by the problem of routing reliably and scalably in a graph, we introduce the notion of a *splicer*, the union of a small number of spanning trees of a graph. We prove that for any bounded-degree n-vertex graph, the union of two uniformly *random* spanning trees approximates the expansion of the graph to within a factor of $O(\log n)$. For the complete graph, we prove that the union of two uniformly random spanning trees is an expander with high probability. For the random graph $G_{n,p}$, for $p = \Omega(\log n/n)$, we give a randomized algorithm for constructing two spanning trees whose union is an expander. A closely related construction, which we call a *selector*, has similar properties. A random selector of a graph is obtained by starting with any spanning tree of the graph and adding a small number of random edges at each vertex.

Key words. random spanning tree, random mapping, sparsification

AMS subject classifications. 68R05, 68R10

DOI. 10.1137/120890971

1. Introduction.

In this paper, we present a new method for obtaining sparse expanders from spanning trees. We begin with some motivation.

Recovery from failures is considered an important problem for the internet today. Ideally, one desires a network where "even right after failure, routing finds [a] path to [the] destination" [23]. How should routing proceed in the presence of link or node failures?

Roughly speaking, to recover from failures, the network should have many alternative paths, a property sometimes called *path diversity*, measured in several ways, including network reliability and congestion. It is well known that expander graphs have low congestion and remain connected even after many (random) failures. Indeed, there is a large literature on routing to minimize congestion and on finding disjoint paths that is closely related to the study of expansion (or more generally, conductance); e.g., [22, 13, 3].

In practice, efficient routing also needs to be compact and *scalable*; in particular, the memory overhead should be linear or sublinear in the number of vertices. This requirement is satisfied by routing using trees, one tree per destination. In fact, the most commonly used method in practice is shortest path routing, which is effectively one tree per destination.[1] Since the final destination determines the next edge to be used, this gives an $O(n)$ bound on the size of the routing table that needs to be stored at each vertex. If a constant-factor stretch is allowed, this can be reduced. For example, with stretch 3, tables of size $O(\sqrt{n})$ suffice as shown by Abraham et al. [1].

The main problem with shortest-path routing or any tree-based scheme is the lack of path diversity. Failing any edge disconnects some pairs of vertices. Recovery

*Received by the editors September 11, 2012; accepted for publication (in revised form) January 9, 2014; published electronically March 25, 2014.

http://www.siam.org/journals/sicomp/43-2/89097.html

†Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (alan@random.math.cmu.edu). The work of this author was partially supported by NSF grant ccf1013110.

‡Microsoft Research India, #9 Lavelle Road, Bangalore, India (navin001@gmail.com).

§Computer Science and Engineering, Ohio State University, Columbus, OH 43210 (Lrademac@cse.ohio-state.edu).

¶School of Computer Science, Georgia Tech, Atlanta, GA 30332 (vempala@math.mit.edu). The work of this author was partially supported by NSF grant AF0915903.

[1]This is called Open Shortest Path First.

is usually achieved by recomputing shortest path trees in the remaining network, an expensive procedure. Moreover, congestion (number of pairs using the same edge) can be high for tree-based routing, despite the fact that the underlying graph might have high expansion, implying that low congestion and high fault-tolerance are possible. There is evidence that AS-level internet topologies are expanders; some stochastic models for networks lead to expanders [16]. However, known algorithms that achieve near-optimal congestion use arbitrary paths in the network and therefore violate the scalability requirement. This raises the following question: is it possible to have a routing scheme that is both scalable and achieves congestion and fault-tolerance approaching that of the underlying graph?

Our work is motivated by the idea and experimental results of the method known as *path splicing* [18, 17], a conceptually simple extension of tree-based routing to multiple trees. With one tree there is a unique path between any two points. With two trees, by allowing a path to switch between the trees multiple times, there could be a large number of available paths. Motiwala et al. showed experimentally that a small number of randomly perturbed shortest path trees for each destination leads to a highly reliable routing method: the union of these trees has reliability approaching that of the underlying graph.

This raises the question whether the results of this experiment can be true in general. In other words, for a given graph does there exist a small collection of spanning trees such that the reliability of the union approaches that of the base graph? As a first step, we study the question of whether for a given graph the union of a few spanning trees captures the expansion of the original graph. In this paper, we propose very simple constructions that use only a small number of trees *in total* (as opposed to one tree per destination) and work for graphs with bounded degrees, random graphs, and the complete graph. The trees are chosen independently from the uniform distribution over all spanning trees, a distribution that can be sampled efficiently with simple algorithms. The simplest of these, due to Aldous [2] and Broder [8], is to take a random walk in the graph and include in the tree every edge that goes to a previously unvisited vertex. Roughly speaking, our main result is that for bounded degree graphs and for the complete graph, a small number of such trees give a subgraph with expansion comparable to the original graph for each cut. Splicers can thus be viewed as a new way to construct expanders with $O(n)$ edges.

In our routing application, the fault-tolerance property we want from splicers is that no cut that was large in the original graph is small in the splicer, and thus we are looking for one-sided approximation of cuts in the original graph. If we look for two-sided approximation, then we get the well-studied notion of *cut-sparsifiers*: The goal here is to approximate every cut using only a small subgraph of the original graph. Note that the property of having efficient routing is not required, unlike for splicers. Cut sparsifiers were first defined by Benczur and Karger [6], who gave an algorithm to construct cut-sparsifiers with $O(n \log n)$ edges. Spielman and Srivastava [24] used a stronger definition requiring that the sparsifier should approximate the Laplacian quadratic form of the original graph. They gave an algorithm that constructs $O(n \log n)$-size sparsifiers. Batson, Spielman, and Srivastava [5] gave an algorithm to construct $O(n)$-size sparsifiers. It is an important problem to find simple and fast sparsification algorithms. Splicers, constructed using random spanning trees obtained by a simple random process called Process B_p, provide cut-sparsifiers of size $O(n)$ for random graphs in $G_{n,p}$: When the base graph is random, with high probability, the union of two spanning trees approximates all cuts to within a factor

of $O(\log n)$. A similar result holds for random selectors. This is a modest step toward a simple and fast algorithm for graph sparsification.

1.1. Our results. A *k-splicer* is the union of k spanning trees of a graph. By a random k-splicer we mean the union of k uniformly randomly chosen spanning trees. We show that for any bounded degree graph, the union of two random spanning trees of the graph approximates the expansion of every cut of the graph. Using more trees gives a better approximation. In the following, $\delta_G(A)$ stands for the set of edges in graph G that have exactly one endpoint in A, a subset of vertices of G.

THEOREM 1.1. *For a graph $G = (V, E)$ with degree at most d, let U_G^k be a random k-splicer, obtained by the union of k uniformly random spanning trees. Also let $\alpha > 0$ be a constant and $\alpha(k-1) \geq 9d^2$. Then with probability $1 - o(1)$, for every $A \subset V$, we have*

$$|\delta_{U_G^k}(A)| \geq \frac{1}{\alpha \log n} \cdot |\delta_G(A)|.$$

Our proof of this makes novel use of a known property of a random spanning tree of a graph, namely, the events of two given edges in the graph being included in the tree are negatively correlated.

Next we show that the factor $1/\log n$ is the best possible for k-splicers constructed from random spanning trees for any constant k. Definitions of expansion and expanders referred to below can be found in section 2.

THEOREM 1.2. *For every n, there is a bounded-degree edge expander G on n vertices such that with probability $1 - o(1)$ the edge expansion of a random k-splicer U_G^k is at most $k^2/C \log n$ for any $k \geq 1$ and a constant $C > 0$ depending only on the maximum degree of G.*

For the complete graph, one can do better, requiring only two trees to get a constant-factor approximation. In fact, we get constant vertex expansion.

THEOREM 1.3. *The union of two uniformly random spanning trees of the complete graph on n vertices has constant vertex expansion with probability $1 - o(1)$.*

Since constant vertex expansion implies constant edge expansion, we get that the union of two uniform random spanning trees has constant edge expansion with high probability.

Next we turn to the random graph $G_{n,p}$. Our main result here is that w.h.p., $G_{n,p}$ has two spanning trees whose union has constant vertex expansion. We give a simple random process (called *Process B_p* henceforth) to find these trees.

THEOREM 1.4. *There exists an absolute constant C, such that for $p \geq C \log n/n$ with probability $1-o(1)$, the union of two random spanning trees obtained from Process B_p applied to a random graph H drawn from $G_{n,p}$ has constant vertex expansion.*

The proof of this theorem is via a coupling lemma (Lemma 7.2) showing that a tree generated by Process B_p applied to a random graph H is nearly uniform among spanning trees of the complete graph.

Theorem 1.4 relates to the work of [6, 24, 5] and leads to simple linear-size sparsifiers with nontrivial approximation guarantees for random graphs. Let $w(\delta_{H'}(A))$ denote the sum of the weights of the edges in $\delta_{H'}(A)$.

THEOREM 1.5. *There exists an absolute constant $C > 0$ such that for $p \geq C \log n/n$ the following holds: Let H be a $G_{n,p}$ random graph and let H' be the 2-splicer obtained from it via process B_p with a weight of pn on every edge. Then with probability $1 - o(1)$, for every $A \subset V$ we have*

$$c_1|\delta_H(A)| \leq w(\delta_{H'}(A)) \leq c_2|\delta_H(A)| \log n$$

for some absolute constants $c_1, c_2 > 0$.

We now provide an alternative construction that gives basically the same results. We will replace our k random trees by the union of (i) an arbitrary tree and (ii) $k-1$ random G-mappings. A G-mapping is a function $f : V \to V$ such that $(v, f(v)) \in E$ for all $v \in V$. Less formally, in a random G-mapping each $v \in V$ independently chooses a uniformly random neighbor $f(v)$. As we are always dealing with loopless graphs, we can assume that $f(v) \neq v$ here. We call this construction a *k-selector*.

When $G = K_n$, it is known that the graph G_f induced by the edges $(v, f(v))$ is "close" to being a spanning tree. G_f is the union of γ cycles containing $\bar{\gamma}$ vertices and a forest of rooted trees with a root for each cycle vertex. Furthermore, $\mathsf{E}(\gamma) \sim \log n$ and $\mathsf{E}(\bar{\gamma}) \sim (\pi n/2)^{1/2}$ (see, e.g., Chapter XIV of Bollobás [7]). Hopefully, this gives the reader some intuition as to why k-selectors have similar properties to k-splicers.

THEOREM 1.6. *For a graph $G = (V, E)$ with vertex degree at most d, let W_G^k be a random k-selector, $k \geq 2$, obtained by the union of an arbitrary tree and $k-1$ independent random G-mappings. Let $\theta_k = 1 - (1 - \frac{1}{d})^{2k-2}$ and let $\alpha = 16/\theta_k$. Then with probability $1 - o(1)$, for every $A \subset V$, we have*

$$|\delta_{W_G^k}(A)| \geq \frac{1}{\alpha \log n} \cdot |\delta_G(A)|.$$

In analogy to Theorem 1.3, we note that the union of k random K_n-mappings is a well-studied model called G_{k-out}.

THEOREM 1.7. *G_{2-out} has constant vertex expansion w.h.p.*

We now consider the random graph $G_{n,p}$.

THEOREM 1.8. *If $p \geq \frac{1+\epsilon}{n} \log n$, where $\epsilon > 0$ is constant, then the following holds: Let G be a $G_{n,p}$ random graph and let H be the graph obtained from it by letting each vertex independently choose two neighbors. Put a weight of pn on every edge. Then with probability $1 - o(1)$, for every $A \subset V$ we have*

$$c_1 |\delta_G(A)| \leq w(\delta_H(A)) \leq c_2 |\delta_G(A)| \log n$$

for some absolute constants $c_1, c_2 > 0$.

Note that w.h.p. $G(n, p)$ is connected and has minimum degree $\Omega(\log n)$ for $p \geq \frac{1+\epsilon}{n} \log n$.

The theorem can be strengthened so that $p = \frac{1}{n}(\log n + \omega(n))$, where $\omega(n) \to \infty$. For slowly growing ω we can have $O(\log n)$ vertices of degree one. This makes the calculations more complicated, and we do not include them here.

1.2. Related work. The idea of using multiple routing trees and switching between them is inspired by the work of [18], which proposed a multipath extension to standard tree-based routing. The method, called path splicing, computes multiple trees to each destination vertex, using simple methods to generate the trees; in one variant, each tree is a shortest path tree computed on a randomly perturbed set of edge weights. Path splicing appears to do extremely well in simulations, approaching the reliability of the underlying graph using only a small number of trees.[2]

Sampling for approximating graph cuts was introduced by Karger, first for global min-cuts and then extended to min s-t cuts and flows. The most recent version due to Benczur and Karger [6] approximates the weight of every cut of the graph within factors of $1+\varepsilon$ and $1-\varepsilon$ using $O(n \log n/\varepsilon^2)$ samples; edges are sampled independently with probability inversely proportional to a connectivity parameter, and each chosen

[2]It has several other features from a practical viewpoint, such as allowing end vertices to specify paths, that we do not discuss in detail here.

edge is weighted with the reciprocal of their probability. Recently, Spielman and Srivastava [24] gave a similar method where edges are sampled independently with probability proportional to the effective resistance and weighted in a similar way, by the reciprocal of the probability with which they are chosen. They show that the quadratic form of the Laplacian of the original graph is approximated within factors $1-\varepsilon$ and $1+\varepsilon$. The similarity in the two methods extends to their analysis also—both parameters, edge strength, and edge resistance share a number of useful properties.

A well-known fact (e.g., [14]) about uniformly random spanning trees is that the probability that an edge e belongs to the uniformly random spanning tree is equal to the effective resistance of e. (There are several equivalent definitions of effective resistance. One of them is the following: Thinking of the graph as an electrical network, let each edge have unit resistance, and then the effective resistance of e is the potential difference applied to the endpoints of e to induce a unit current.) This fact shows a connection of our work with [24], which samples edges in a graph according to their effective resistances to construct a sparsifier.

It is well known that the union of three random perfect matchings in a complete graph with an even number of vertices (see, e.g., [11]) is an expander with high probability. Our result on the union of random spanning trees of the complete graph can be considered as a result in a similar vein. While our proof for spanning trees has a similar high-level outline, it seems to require new ideas. On the other hand, our result for the union of spanning trees of bounded degree graphs does not seem to have any analogue for the union of matchings. Indeed, generating random perfect matchings of graphs is a highly nontrivial problem—computing the permanent of 0–1 matrices being the special case of bipartite graphs [12].

2. Preliminaries. Let $G = (V, E)$ be an undirected graph. For $v \in V$ define $\Gamma(v) := \{u \in V : (u, v) \in E\}$, the set of neighbors of v. For $A \subseteq V$, define $\Gamma(A) := \cup_{v \in A}\Gamma(v)$ and $\Gamma'(A) := \Gamma(A) \setminus A$. Finally, let $\delta_G(A) := \{(u, v) \in E : u \in A, v \notin A\}$. The edge expansion of G is

$$\min_{A \subseteq V, 1 \le |A| \le |V|/2} \frac{|\delta_G(A)|}{|A|}.$$

The vertex expansion of G is

$$\min_{A \subseteq V, 1 \le |A| \le |V|/2} \frac{|\Gamma'(A)|}{|A|}.$$

We say that a family of graphs is an edge expander (family) if the edge expansion of the family is bounded below by a positive constant. Vertex expanders are defined similarly.

Let K_n denote the complete graph on n vertices.

For $a \in \mathbb{R}$, let $[a] := \{i \in \mathbb{N} : 1 \le i \le a\}$. On several occasions we will use the inequality $\binom{n}{k} \le (\frac{ne}{k})^k$.

3. Uniform random spanning trees. Uniformly random spanning trees of graphs are fairly well-studied objects; see, e.g., [15]. In this section we describe properties of random spanning trees that will be useful for us. There are several algorithms known for generating a uniformly random spanning tree of a graph, e.g., [2, 8, 21, 15]. The algorithm due to Aldous and Broder is very simple and will be useful in our analysis: Start a uniform random walk at some arbitrary vertex of the graph, and when the walk visits a vertex for the first time, include the edge used

to reach that vertex in the tree. When all the vertices have been visited, we have a spanning tree which is uniformly random *regardless of the initial vertex*.

For a connected base graph $G = (V, E)$, random variable T_G denotes a uniformly random spanning tree of G. U_G^k will denote the union of k such trees chosen independently. For edge $e \in E$, abusing notation a little, we will refer to events $e \in E(T_G)$ and $e \in E(U_G^k)$ as $e \in T_G$ and $e \in U_G^k$.

Negative correlation of edges. The events of various edges belonging to the random spanning tree are negatively correlated: For any subset of edges $e_1, \ldots, e_k \in E$, we have

$$\mathsf{P}\big[(e_1 \in T_G) \wedge (e_2 \in T_G) \wedge \cdots \wedge (e_k \in T_G)\big] \;\leq\; \mathsf{P}[e_1 \in T_G]\,\mathsf{P}[e_2 \in T_G] \cdots \mathsf{P}[e_k \in T_G]. \tag{3.1}$$

A similar property holds for the complementary events:

$$\mathsf{P}\big[(e_1 \notin T_G) \wedge (e_2 \notin T_G) \wedge \cdots \wedge (e_k \notin T_G)\big] \;\leq\; \mathsf{P}[e_1 \notin T_G]\,\mathsf{P}[e_2 \notin T_G] \cdots \mathsf{P}[e_k \notin T_G]. \tag{3.2}$$

These are easy corollaries of [15, Theorem 4.5], which in turn is based on the work of Feder and Mihail [10] and a classical result that (3.1) holds for two edges.

Negatively correlated random variables and tail bounds. For $e \in E$, define indicator random variables X_e to be 1 if $e \in T$, and 0 otherwise. Then we can rewrite (3.1) as follows.

For any subset of edges $e_1, \ldots, e_k \in E$, we have

$$\mathsf{E}[X_{e_1} \cdots X_{e_k}] \leq \mathsf{E}[X_{e_1}] \cdots \mathsf{E}[X_{e_k}]. \tag{3.3}$$

For random variables $\{X_e\}$ satisfying (3.3), we say that $\{X_e\}$ are negatively correlated. Several closely related notions exist; see Dubhashi and Ranjan [9] and Pemantle [20]. Dubhashi and Ranjan [9] gave a property of negative correlation that will be useful for us: It essentially says that Chernoff's bound for the tail probability for sums of independent random variables applies unaltered to negatively correlated random variables. More precisely, we will use the following version of Chernoff's bound.

THEOREM 3.1. *Let $\{X_i\}_{i=1}^n$ be a family of* 0–1 *negatively correlated random variables such that $\{1 - X_i\}_{i=1}^n$ are also negatively correlated. Let p_i be the probability that $X_i = 1$. Let $p := \frac{1}{n}\sum_{i\in[n]} p_i$. Then for $\lambda > 0$,*

$$\mathsf{P}\left[\sum_{i\in[n]} X_i < pn - \lambda\right] \leq e^{-\lambda^2/(2pn)}.$$

Proof. The proof splits into two steps: In the first step we prove that for arbitrary λ, we have

$$\mathsf{E}\left[\exp\left(\lambda \sum_{i=1}^n X_i\right)\right] \leq \prod_{i=1}^n \mathsf{E}\big[\exp(\lambda X_i)\big]. \tag{3.4}$$

The second step is a standard Chernoff bound argument as in the proof of Theorem A.1.13 in [4]. Since the first step is short and perhaps not that well known,

we provide a proof here for completeness. In this, we basically follow Dubhashi and Ranjan [9].

The case $\lambda = 0$ is trivially true. We now prove (3.4) for $\lambda > 0$. Since X_i's take 0–1 values, for any integers $a_1, \ldots, a_n > 0$, we have $X_1^{a_1} X_2^{a_2} \cdots X_n^{a_n} = X_1 X_2 \cdots X_n$. Now, writing $\exp(\lambda \sum_{i=1}^n X_i)$ using the Taylor series for e^x, and expanding each summand, we get a sum over various monomials over the X_i's. For each monomial we have by the definition of negative correlation that $\mathsf{E}[X_1 \cdots X_n] \leq \prod_{i=1}^n \mathsf{E}[X_i]$. This gives (3.4) for $\lambda > 0$.

For $\lambda < 0$, a similar argument using $1 - X_i$ in the role of X_i gives (3.4). □

4. Expansion when base graph is a complete graph. Our proof here has the same high-level outline as the proof for showing that the union of three random perfect matchings in a complete graph with an even number of vertices is a vertex-expander (see, e.g., [11]): One shows that for any given vertex set A of size $\leq n/2$, the probability is very small for the event that $|\Gamma'(A)|$ is small in the union of the matchings. A union bound argument then shows that the probability is small for the existence of any set A with $|\Gamma'(A)|$ small. However, new ideas are needed in our case because spanning trees are generated by the random walk process, which appears to be more complex to analyze than random matchings in complete graphs.

Proof of Theorem 1.3. For given $A \subseteq V$, $|A| = a$, and given expansion constant c, we will upper bound the probability that $|\Gamma'_T(A)| \leq ca$ for random spanning tree T in K_n. To this end, we fix a set $A' \subseteq V \setminus A$ of size $\lfloor ca \rfloor$ and bound the probability that $\Gamma'_T(A) \subseteq A'$, and then we use a union bound over all possible choices of A and A' to show that no such A, A' are likely to exist. Without loss of generality the vertices are labeled $V = \{1, \ldots, n\}$, $A = [a] = \{1, \ldots, a\}$, and $A' = \{a+1, a+\lfloor ca \rfloor\}$. The probability that there exists a set $A \subseteq V$ such that $|A| \leq n/2$ and $|\Gamma'_T(A)| \leq ca$ in the union of t random independent spanning trees is at most

$$\sum_{a=1}^{\lfloor n/2 \rfloor} \binom{n}{a} \binom{n}{\lfloor ca \rfloor} \mathsf{P}\big(\Gamma_T(A) \subseteq [a + ca]\big)^t. \tag{4.1}$$

We divide the sum into two parts and bound them separately: For $a \leq n/12$, we use the random walk construction of the random spanning tree which, as we will see, can be interpreted as every vertex in A essentially picking a random neighbor (but not in a completely independent way). For $a \in (n/12, n/2]$, we look at all the edges of the cut as if they were independently selected in the spanning tree by negative correlation.

For $a \leq n/12$, we first consider a random walk on V starting outside A. We use this walk to generate the random spanning tree. Let $X_1, X_2, \ldots$ denote the states of this random walk. Let $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_a\}$ be the times when the walk adds a new member of A to the tree. Let U_t denote the set of vertices in A that have not been visited at the completion of step t.

At the completion of step τ_i, we have $X_{\tau_i} \notin U_{\tau_i}$ and $|U_{\tau_i}| = a - i$ and so

$$\mathsf{P}(X_{\tau_i+1} \in U_{\tau_i}) = \frac{a-i}{n-1}. \tag{4.2}$$

This holds conditional on $X_1, X_2, \ldots, X_{\tau_i}$.

In addition, if $B = [a + ca]$, then

$$\mathsf{P}(X_t \in [a+ca] \mid U_t, t \notin \mathcal{T}, t+1 \in \mathcal{T}) \leq \frac{|B| - |U_t|}{n - 1 - |U_t|}. \tag{4.3}$$

Here the conditioning tells us that X_t is chosen uniformly in $B \setminus (U_{t-1} \cup \{X_{t-1}\})$, whereas X_t has $n-1-|U_{t-1}|$ choices overall. Note that $U_t = U_{t-1}$. Also, (4.3) holds conditional on $X_1, X_2, \ldots, X_{t-1}$.

Let k be the number of times that $X_{\tau_i+1} \in U_{\tau_i}$. Then

$$\begin{aligned}
\mathsf{P}(\Gamma_T(A) \subseteq [a+ca]) &\le \sum_{k=0}^{a-1} \left(\prod_{j=1}^{a-k} \frac{a+ca-j}{n-1-j} \right) \binom{a-1}{k} \left(\prod_{i=1}^{k} \frac{a-i}{n-1} \right) \\
&\le \sum_{k=0}^{a-1} \binom{a-1}{k} \left(\frac{a+ca}{n-1} \right)^{a-k} \left(\frac{a}{n} \right)^{k} \\
&= \frac{a+ca}{n} \left(\frac{a+ca}{n-1} + \frac{a}{n} \right)^{a-1} \\
&\le \left(\frac{2(1+c)a}{n} \right)^{a}.
\end{aligned} \tag{4.4}$$

Explanation of (4.4). If we fix k, then $\binom{a-1}{k}$ determines the i for which $X_{\tau_i+1} \in U_{\tau_i}$. The product terms maximize the corresponding products (4.2), (4.3) under these circumstances.

We now use this in (4.1) for $a \le n/12$. Let $K = 2(1+c)$.

$$\begin{aligned}
&\sum_{a=1}^{\lfloor n/12 \rfloor} \binom{n}{a} \binom{n}{\lfloor ca \rfloor} \mathsf{P}\big(\Gamma_T(A) \subseteq [a+ca]\big)^t \\
&\le \sum_{a=1}^{\lfloor n/12 \rfloor} \left(\frac{en}{a} \right)^{a} \left(\frac{en}{ca} \right)^{ca} \left(\frac{aK}{n} \right)^{at} \\
&= \sum_{a=1}^{\lfloor n/12 \rfloor} \alpha^a K^{at} \left(\frac{a}{n} \right)^{a(t-1-c)} \qquad \left(\text{where } \alpha = \frac{e^{1+c}}{c^c} \right) \\
&\le \sum_{a=1}^{\lfloor \sqrt{n} \rfloor} \alpha^a K^{at} \left(\frac{1}{\sqrt{n}} \right)^{a(t-1-c)} + \sum_{a=\lfloor \sqrt{n} \rfloor + 1}^{\lfloor n/12 \rfloor} \alpha^a K^{at} \left(\frac{1}{12} \right)^{a(t-1-c)} \\
&\le \left[\alpha K^t n^{-(t-1-c)/2} + \left(\frac{\alpha K^t}{12^{t-1-c}} \right)^{\lfloor \sqrt{n} \rfloor + 1} \right] \frac{1}{1 - \alpha K^t 12^{-(t-1-c)}},
\end{aligned}$$

which goes to 0 as $n \to \infty$ when $\alpha K^t / 12^{t-1-c} < 1$, and this happens for $t = 2$ and a sufficiently small constant c.

For the rest of the sum in (4.1), $a \in (n/12, n/2]$, we use negative correlation of the edges of the random spanning tree T (section 3) to estimate the probability that $\Gamma_T(A) \subseteq [a+ca]$. Any fixed edge from K_n appears in T with probability $2/n$. We have that $\Gamma_T(A) \subseteq [a+ca]$ iff no edge between A and $V \setminus [a+ca]$ is present in T, and negative correlation (3.2) implies that this happens with probability at most

$(1-2/n)^{a(n-(a+ca))}$. Thus,

$$\begin{aligned}
&\sum_{a=\lfloor n/12\rfloor+1}^{\lfloor n/2\rfloor}\binom{n}{a}\binom{n}{\lfloor ca\rfloor}\mathsf{P}(\Gamma_T(A)\subseteq[a+ca])^t\\
&\quad\le\sum_{a=\lfloor n/12\rfloor+1}^{\lfloor n/2\rfloor}\left(\frac{en}{a}\right)^a\left(\frac{en}{ca}\right)^{ca}\left(1-\frac{2}{n}\right)^{ta(n-(a+ca))}\\
&\quad\le n\sup_{\gamma\in[1/12,1/2]}\left(\frac{e}{\gamma}\right)^{\gamma n}\left(\frac{e}{c\gamma}\right)^{c\gamma n}\left(1-\frac{2}{n}\right)^{t\gamma n(n-(1+c)\gamma n))}\\
&\quad\le n\sup_{\gamma\in[1/12,1/2]}\left(\frac{(e/\gamma)^{1+c}}{c^c}\right)^{\gamma n}e^{-2t\gamma n(1-(1+c)\gamma)}\\
&\quad= n\sup_{\gamma\in[1/12,1/2]}\left(\frac{(e/\gamma)^{1+c}}{c^ce^{2t(1-(1+c)\gamma)}}\right)^{\gamma n}.
\end{aligned}$$

For any fixed $c>0$, the function

$$f(\gamma)=\frac{(e/\gamma)^{1+c}}{c^ce^{2t(1-(1+c)\gamma)}}$$

is convex for $\gamma>0$ and hence the sup is attained at one of the boundary points $1/12$ and $1/2$, and the function is strictly less than 1 at these boundary points for $t=2$ and a sufficiently small constant c. This implies that this sum goes to 0 as $n\to\infty$. □

5. Expansion when base graph is a bounded-degree graph: positive result. In this section we consider graphs with bounded degrees. To simplify the presentation we restrict ourselves to regular graphs; it is easy to drop this restriction at the cost of extra notation. We show that for constant degree graphs the edge expansion is captured fairly well by the union of a small number of random spanning trees.

Proof of Theorem 1.1. It follows by the random walk construction of random spanning trees that for any edge $(u,v)\in E$, we have $\mathsf{P}[(u,v)\in T]\ge 1/d(u)$. To see this, note that if we start the random walk at vertex u, then with probability $1/d(u)$ the first traversed edge is (u,v), which then gets included in T. Thus for $A\subset V$, we have that

$$\mathsf{E}\big[|\delta_{T_G}(A)|\big]\ge\frac{1}{d}\cdot|\delta_G(A)|.$$

We would now like to use the above expectation result to prove our theorem. Recall the definition of random variables X_e from section 3: For edge $e\in E$, X_e is the indicator random variable taking value 1 if $e\in T$, and value 0 otherwise. Thus we have $|\delta_T(A)|=\sum_{e\in\delta_G(A)}X_e$. We want to show that $\sum_{e\in\delta_G(A)}X_e$ is not much smaller than its expectation with high probability. Random variables X_e are not independent. Fortunately, they are negatively correlated as we saw in section 3, which allows us to use Theorem 3.1:

$$\mathsf{P}\left[\sum_{e\in\delta_G(A)}X_e<p|\delta_G(A)|-\lambda\right]<e^{-\lambda^2/(2p|\delta_G(A)|)}\le e^{-\lambda^2/(2|\delta_G(A)|)},\tag{5.1}$$

where p is the average of $\mathsf{P}[X_e = 1]$ for $e \in \delta_G(A)$. Since $\mathsf{P}[X_e = 1] \geq 1/d$ for all edges e, we have $p \geq 1/d$, and for $\lambda = (p - 1/(2d))|\delta_G(A)|$ we have

$$\mathsf{P}\left[|\delta_{T_G}(A)| < \frac{1}{2d}|\delta_G(A)|\right] < e^{-\frac{|\delta_G(A)|}{8d^2}}.$$

This gives

$$\mathsf{P}\left[|\delta_{U_G^k}(A)| < \frac{1}{2d}|\delta_G(A)|\right] < e^{-\frac{k|\delta_G(A)|}{8d^2}}. \tag{5.2}$$

Now we estimate the probability that there is a *bad cut*, namely, a cut A such that $|\delta_{U_G^k}(A)| = a$ and $|\delta_G(A)| \geq \alpha a \ln n$. To do this we first look at cuts of size a in the first random tree, which have size at least $\alpha a \ln n$ in G. (This step is necessary: the modified Chernoff bound that we use is only as strong as the independent case, and when edges are chosen independently one is likely to get isolated vertices; looking at the first tree ensures that this does not happen.) In order to be bad, these cuts must have small size in all the remaining trees. The probability of that happening is given by (5.2). The number of cuts in the first tree of size a is clearly no more than $\binom{n-1}{a} < \binom{n}{a}$, as there are $\binom{n-1}{a}$ ways of picking a edges out of $n-1$, although not all of these may correspond to valid cuts. Then, the probability that a bad cut exists is at most

$$\begin{aligned}
\sum_{a=1}^{n/\ln n} \binom{n}{a} e^{-\frac{(k-1)\alpha a \ln n}{8d^2}} &\leq \sum_{a=1}^{n/\ln n} \left(\frac{en}{a}\right)^a e^{-\frac{(k-1)\alpha a \ln n}{8d^2}} \\
&= \sum_{a=1}^{n/\ln n} \exp\left(\left(\ln(en/a) - \frac{(k-1)\alpha \ln n}{8d^2}\right) a\right) \\
&= \sum_{a=1}^{n/\ln n} \exp\left(\left(\ln(e/a) + \left(1 - \frac{(k-1)\alpha}{8d^2}\right)\ln n\right) a\right).
\end{aligned}$$

Choosing $(k-1)\alpha > 9d^2$ makes the above sum $o(1)$. ∎

6. Expansion when base graph is a bounded-degree graph: Negative result. Here we show that Theorem 1.1 is the best possible up to a constant factor for expansion.

Proof of Theorem 1.2. We begin with a d-regular edge expander G' on n vertices with a Hamiltonian cycle, where $d > 4$ is a fixed integer. (It is easy to construct such expanders by starting with a $(d-2)$-regular expander and adding edges of a Hamiltonian cycle to it, so that the graph can be completed to a d-regular graph. We omit the easy details.) Let $0 < \ell < \log n$ be an integer to be chosen later, and let H be a Hamiltonian path in G'. Subdivide H into subpaths $P_1, \ldots, P_{n/\ell}$, each of length ℓ. (To keep the formulas simple, we suppress the integrality issues here which are easily taken care of.)

For two subpaths P_i and P_j, we say that they *interact* if $(P_i \cup \Gamma'(P_i)) \cap (P_j \cup \Gamma'(P_j)) \neq \emptyset$. Since G' is d-regular, $|\Gamma'(P_i)| \leq d\ell$. So, any subpath can interact with

at most $d^2\ell$ other subpaths. Indeed, P_i can interact with P_j only if there is a vertex in P_j within distance two of a vertex in P_i. There are at most $d^2\ell$ vertices at distance two from a vertex of P_i, and as the P_r's are vertex disjoint, each one of these vertices within distance two of a vertex in P_i can be in at most one such P_j. So there are at most $d^2\ell$ P_j's that interact with any given P_i.

Thus we can find a set I of $\frac{1}{d^2\ell} \cdot n/\ell$ paths among $P_1, \ldots, P_{n/\ell}$, so that no two paths in I interact.

We now describe the construction of G, which will be obtained by adding edges to G'. For each path $P \in I$, we do the following. Add an edge between the two end-points of P if such an edge did not already exist in G'. If the subgraph $G[\Gamma'(P)]$ induced by the neighborhood of path P does not have a Hamiltonian cycle, then we add edges to it so that it becomes Hamiltonian. Clearly, in doing so we only need to increase the degree of each vertex by at most two. The final graph that we are left with is our G. For each path $P \in I$, we fix a Hamiltonian cycle in $G[\Gamma'(P)]$, and we also have the cycle of which P is a part. We denote these two cycles by $C_1(P)$ and $C_2(P)$.

We will generate a random spanning tree T of G by the random walk algorithm starting the random walk at some vertex outside of all paths in I. For $P \in I$, we say that event E_P (over the choice of a random spanning tree T of G) occurs if the random walk, on first visit to $C_1(P) \cup C_2(P)$, first goes around $C_1(P)$ without going out or visiting any vertex twice, and then it goes on to traverse $C_2(P)$, again without going out or visiting any vertex twice until it has visited all vertices in $C_2(P)$. For all $P \in I$, we have

$$\mathsf{P}[E_P] \geq 1/(d+2)^{|C_1(P)|+|C_2(P)|-1} \geq 1/(d+2)^{(d+1)\ell-1}. \tag{6.1}$$

If event E_P happens, then in the resulting tree T we have $|\delta_T(V(P))| = 1$. Thus our goal will be to show that with substantial probability there is a $P \in I$ such that E_P happens. Since no two paths in I interact with each other, events E_P are mutually independent. If we are choosing k random spanning trees, then define E_P^k to be the event that E_P occurs for all k spanning trees. Clearly, $\mathsf{P}[E_P^k] = \mathsf{P}[E_P]^k$. Then the probability that E_P^k doesn't occur for any $P \in I$ is at most

$$\begin{aligned}\left(1 - \frac{1}{(d+2)^{k(d+1)\ell-k}}\right)^{|I|} &= \left(1 - \frac{1}{(d+2)^{k(d+1)\ell-k}}\right)^{\frac{n}{d^2\ell^2}} \\ &\leq \exp\left(-\frac{n}{(d+2)^{k(d+1)\ell-k+2}\ell^2}\right).\end{aligned}$$

It follows readily that there is a constant C (that depends on d) such that for $\ell k \leq C \log n$ the above probability is $o(1)$. Hence, with probability $1 - o(1)$ there is a path $P \in I$ such that $|\delta_{U_G^k}(V(P))| \leq k$. The edge expansion of P therefore is $k/\ell = k^2/(C \log n)$ for $\ell = C(\log n)/k$. □

7. Splicers of random graphs. We will construct a random process on random graphs that generates random spanning trees with a distribution that is very close to the uniform distribution on the complete graph. The process first directs edges to mimic the distribution of a directed random graph.

Given an undirected graph H and a parameter $0 < p \leq 1$, construct a random directed graph denoted $D_p(H)$ with vertex set $V(H)$ and independently for every edge (u, v) of H:

- directed edges (u, v) and (v, u) with probability $\frac{-p-2\sqrt{1-p}+2}{p}$,
- only directed edge (u, v) with probability $\frac{p+\sqrt{1-p}-1}{p}$, and
- only directed edge (v, u) with probability $\frac{p+\sqrt{1-p}-1}{p}$.

If H is random according to $G_{n,p}$, then $D_p(H)$ is random with each edge picked with probability $q = 1 - \sqrt{1-p}$. Note that $p/2 \le q \le p$.

Let T be the uniform distribution on spanning trees of K_n. We now describe Process B_p, which is a random process that given an undirected graph H and a parameter $0 < p \le 1$ generates a spanning tree with a distribution that we denote $T_{p,H}$. Consider the following random process that generates a walk in $D_p(H)$ or stops with no output:

1. Start at a vertex v_0 of $D_p(H)$.
2. At a vertex v, an edge is traversed as follows. Suppose $d_1(v)$ out of $d(v)$ outgoing edges at v are previously traversed. Then, the probability of picking a previously traversed edge is $1/(n-1)$ while the probability for each new edge is
$$\frac{1 - \frac{d_1(v)}{n-1}}{d(v) - d_1(v)}.$$
3. If all vertices have been visited, output the walk and stop. If this has not happened and at the current vertex v one has $d_1(v) = d(v)$, stop with no output.

As in the random walk algorithm, the spanning tree given by Process B_p (if it succeeds in visiting all the vertices) is the set of edges that are used on first visits to each vertex, but the random sequence of edges is different here.

A covering walk of a graph is a walk passing through all vertices. Let D be the distribution on covering walks of the (undirected) complete graph starting at a vertex v_0 where a walk is generated by a random walk that starts at v_0 and walks until it has visited all the vertices. Let D_p be the distribution on covering walks of the complete graph given by first choosing H according to $G_{n,p}$ and running Process B_p starting from v_0.

LEMMA 7.1. *There exists an absolute constant c such that for $p > c \log n/n$ the total variation distance*[3] *between the distributions D and D_p is $o(1)$.*

Proof. We will couple D and D_p so that the walk in D picks the same edges as the walk in D_p, but if D_p fails, then D continues its random walk. Then these covering walks coincide whenever D_p succeeds, and thus the probability of success is an upper bound to the total variation distance between D and D_p. Now, D_p does not fail if every vertex in H_d has out-degree at least $c_1 \log n$ and Process B_p does not visit any vertex more than $c_2 \log n$ times for $c_1 > c_2$. A Chernoff bound gives c (from the statement of the lemma) and c_1 such that the first part happens with probability $1 - o(1)$. For the second part, we observe that if there is no failure, then Process B_p behaves exactly like a random walk in the complete graph, and therefore it visits all vertices in at most $c_3 n \log n$ steps with probability $1 - o(1)$ for some constant c_3 (this is essentially the coupon collector's problem with $n-1$ coupons; see [19, section 3.6 and Chapter 6]) and a walk of that length does not visit any vertex more than $c_2 \log n$ times with probability $1 - o(1)$ for some constant c_2 (by a straightforward variation of the occupancy problem in [19, section 3.1]). ∎

[3]The *variation distance* $||D_1 - D_2||$ between two distributions on a finite set X is defined as $\frac{1}{2}\sum_{x \in X} |D_1(x) - D_2(x)|$.

Let T_p be the distribution on trees obtained by first choosing H from $G_{n,p}$ and then generating a random spanning tree according to Process B_p.

LEMMA 7.2. *There exists an absolute constant c such that for $p > c \log n/n$, the total variation distance between the distributions T and T_p is $o(1)$.*

Proof. This is immediate from Lemma 7.1, as random trees from T or T_p are just functions of walks from D or D_p, respectively. □

Proof of Theorem 1.4. In the random graph H, we generate two random trees by using one long sequence of edges, with a breakpoint whenever we complete the generation of a spanning tree. In the complete graph, also, we generate two trees from such a sequence obtained from the uniform random walk. Using the same coupling as in Lemma 7.2, we see that these distributions on these sequences have variation distance $o(1)$. Therefore the spanning trees of H obtained by the first process have total variation distance $o(1)$ to random spanning trees of the complete graph. By Theorem 1.3, the union of these trees has constant expansion with probability $1-o(1)$ overall. □

With these results we are ready to prove our theorem about sparsifiers of random graphs:

Proof of Theorem 1.5. We need the fact that for sufficiently large constant C with probability $1 - o(1)$, all cuts $\delta_H(A)$ in random graph H satisfy

$$c_3 p|A|(n - |A|) \le |\delta_H(A)| \le c_4 p|A|(n - |A|). \tag{7.1}$$

This is well known and follows immediately from appropriate Chernoff-type bounds.

We only need to prove the theorem for $|A| \le n/2$. We now prove the first inequality in the statement of the theorem. By Theorem 1.3 with probability $1-o(1)$, for any $A \subset V$ such that $|A| \le n/2$, we have $|\delta_{H'}(A)| \ge c_5|A|$ for some $c_5 > 0$, and so $w(\delta_{H'}(A)) \ge c_5|A|pn \ge c_5 p|A|(n - |A|) \ge \frac{c_5}{c_4}|\delta_H(A)|$.

For the second inequality in the statement of the theorem, we use the fact that the maximum degree of a vertex in a random spanning tree in the complete graph is $O(\log n)$ with probability $1 - o(1)$. Thus, by Lemma 7.2 the same holds for random spanning trees generated by process B_p. We then have $|\delta_{H'}(A)| \le c_6|A| \log n$ for some $c_6 > 0$, and so $w(\delta_{H'}(A)) \le c_6 pn|A| \log n \le 2c_6 pn|A|(n - |A|) \log n \le (2c_6/c_3)|\delta_H(A)| \log n$. □

8. Selectors.

8.1. Expansion when base graph is a bounded-degree graph.

Proof of Theorem 1.6. Let T be an arbitrary spanning tree of G and let $M_1, M_2, \ldots, M_{k-1}$ be independently chosen random G-mappings. Let $H = T \cup M_1 \cup \cdots \cup M_{k-1}$. Let A be a subset of V. Let $a = |\delta_T(A)|$. Since $H \supseteq T$ we can assume now that $|\delta_G(A)| > a\alpha \log n$.

For each edge $e = (u, v) \in \delta_G(A)$ let $\eta_e = 1$ if $\exists i : M_i(u) = v$ or $\exists i : M_i(v) = u$ and zero otherwise and let $X = \sum_{e \in \delta_G(A)} \eta_e$.

Now

$$\mathsf{E}(X) \ge \theta_k |\delta_G(A)|,$$

where

$$\theta_k = 1 - \left(1 - \frac{1}{d}\right)^{2k-2}.$$

Now we will see that X is the sum of negatively correlated $\{0,1\}$ random variables and so we can use Chernoff-type bounds. Therefore

$$\mathsf{P}\left(X \le \frac{1}{2}\,\mathsf{E}(X)\right) \le e^{-\,\mathsf{E}(X)/8} \le e^{-\theta_k|\delta_G(A)|/8} \le n^{-\alpha\theta_k a/8} \le n^{-2a}$$

since $\alpha\theta_k \ge 16$.

The theorem now follows from the fact, observed in section 5, that there are at most $\binom{n-1}{a}$ sets such that $a = \delta_T(A)$.

Negative correlation of the η_e's. Let $e_i = (u_i, v_i) \in \delta_G(A), i = 1, 2\ldots, m$. We have to prove (see (3.3)) that if $e_m \notin \{e_1, e_2, \ldots, e_{m-1}\}$, then

$$\mathsf{P}(\eta_{e_m} = 1 \mid \eta_{e_1} = \cdots = \eta_{e_{m-1}} = 1) \le \mathsf{P}(\eta_{e_m} = 1). \tag{8.1}$$

Because η_{e_m} is independent of η_{e_i} when $e_i \cap e_m = \emptyset$, we will assume that $e_i \cap e_m \neq \emptyset$ for $i = 1, 2, \ldots, m-1$.

Let $\Omega = \{G-\text{mappings}\}^{k-1}$. Partition Ω into $\Omega_1, \Omega_2, \ldots$, where Ω_j is determined by $M_l(u)$, $l = 1, 2, \ldots, k-1$ for $u \notin \{u_m, v_m\}$. Suppose that there are Π parts in all to this partition. Then each Ω_j can be expressed as $\Omega_j = \{\omega_j\} \times N_{u_m}^{k-1} \times N_{v_m}^{k-1}$, where ω_j defines $M_l(u)$, $l = 1, 2, \ldots, k-1$ for $u \notin \{u_m, v_m\}$ and N_{u_m}, N_{v_m} are the G-neighborhoods of u_m, v_m, respectively. Now let $\Omega_j^* = \Omega_j \cap \{\eta_{e_1} = \cdots = \eta_{e_{m-1}} = 1\}$ for $j = 1, 2, \ldots, \Pi$. Equation (8.1) will follow from

$$\mathsf{P}(\eta_{e_m} = 1 \mid \Omega_j^*) \le \mathsf{P}(\eta_{e_m} = 1), \qquad j = 1, 2, \ldots, \Pi. \tag{8.2}$$

(To verify (8.1), we only need to prove this when $\Omega_j^* \neq \emptyset$.)

Next let $I_j = \{i \in [m-1] : \eta_{e_i} = 1 \text{ for all } \omega \in \Omega_j\}$ be those indices i for which $\eta_{e_i} = 1$ is already determined by the choices in ω_j. Here we have $e_i = \{x, y\}$, where $y \in e_m$ and in Ω_j we have $M_l(x) = y$. In this case, η_{e_m} is (conditionally) independent of η_{e_i}. We can therefore assume that $I_j = \emptyset$.

Claim (8.2) now amounts to the following: We randomly place $k-1$ balls into boxes $B_1, B_2, \ldots, B_d$ (the random values of $M_l(u_m)$, $l = 1, 2, \ldots, k-1$) and randomly place $k-1$ balls into boxes $C_1, C_2, \ldots, C_d$ (the random values of $M_l(v_m)$, $l = 1, 2, \ldots, k-1$). (Strictly speaking there are at most d boxes here, but this only causes a change in notation.) Let X_r be the number of balls in box B_r and let Y_s be the number of balls in box C_s. Then if B_1, C_1 correspond to u_m, v_m, then we have to show that

$$\mathsf{P}(X_1 + Y_1 \ge 1 \mid X_l \ge 1, l \in I \text{ and } Y_l \ge 1, l \in J) \le \mathsf{P}(X_1 + Y_1 \ge 1), \tag{8.3}$$

where I, J are subsets of $[d] \setminus \{1\}$.

Now let

$$\pi_1 = \mathsf{P}(X_1 \ge 1 \mid X_l \ge 1, l \in I) \le \pi_1' = \mathsf{P}(X_1 \ge 1),$$
$$\pi_2 = \mathsf{P}(y_1 \ge 1 \mid Y_l \ge 1, l \in I) \le \pi_2' = \mathsf{P}(Y_1 \ge 1),$$

where the inequalities follow from [9], Theorem 13.

The independence of the X_i's and Y_i's implies that

$$LHS(8.3) = 1 - (1-\pi_1)(1-\pi_2) \le 1 - (1-\pi_1')(1-\pi_2') = RHS(8.3). \qquad \square$$

8.2. Expansion when base graph is a complete graph.

Proof of Theorem 1.7. Let $G = G_{2-out}$. We choose some small positive ϵ. Then the probability that G has vertex expansion at most ϵ can be bounded by

$$\sum_{k=3}^{n/2} \binom{n}{k}\binom{n}{\epsilon k}\left(\frac{k+\epsilon k}{n}\right)^{2k}\left(\frac{n-k}{n}\right)^{2(n-k-\epsilon k)}. \tag{8.4}$$

The sum starts at 3 because in G_{2-out} every set of size one or two has at least one neighbor:

$$\begin{aligned}&\le \sum_{k=3}^{n/2}\left(\frac{ne}{k}\right)^{k}\left(\frac{ne}{\epsilon k}\right)^{\epsilon k}\left(\frac{k+\epsilon k}{n}\right)^{2k}\left(\frac{n-k}{n}\right)^{2(n-k-\epsilon k)}\\ &= \sum_{k=3}^{n/2} u_k,\end{aligned}$$

where

$$u_k = \left(\left(\frac{k}{n}\right)^{1-\epsilon}\cdot e\cdot\left(\frac{e}{\epsilon}\right)^{\epsilon}\cdot(1+\epsilon)^2\right)^{k}\left(\frac{n-k}{n}\right)^{2(n-k-\epsilon k)}.$$

When $k \le n^{1/2}$, $u_k \le v_k = (\left(\frac{k}{n}\right)^{1-\epsilon}\cdot e\cdot\left(\frac{e}{\epsilon}\right)^{\epsilon}\cdot(1+\epsilon)^2)^k \le n^{-k/3}$. If $k \le n/3$, then $v_k \le (0.95)^k$ for ϵ sufficiently small. If $n/3 \le k \le n/2$, then $\left(\frac{n-k}{n}\right)^{2(n-k-\epsilon k)} \le \left(\frac{2}{3}\right)^{2(1-\epsilon)k}$ and so $u_k \le (\frac{e^{1+\epsilon}(1+\epsilon)^2}{2^{1-\epsilon}\epsilon^{\epsilon}}\cdot\left(\frac{2}{3}\right)^{2(1-\epsilon)})^k \le (0.7)^k$ for ϵ sufficiently small. □

8.3. Selectors of random graphs.

Proof of Theorem 1.8. The upper bound proof of Theorem 1.5 rests on the fact that the maximum degree of a random tree is $O(\log n)$ w.h.p. This is also true for a random mapping (actually $o(\log n)$ to be precise), and so the proof of the upper bound goes through unchanged.

For the lower bound we let $M_1 = k+\epsilon k-1$ and $N_1 = n-M_1$ and $M_2 = n-k-1$ and $N_2 = n-M_2$ and replace (8.4) by

$$o(1) + \sum_{k=3}^{n/2}\binom{n}{k}\binom{n}{\epsilon k}A_1^k A_2^{n-k-\epsilon k}. \tag{8.5}$$

The $o(1)$ term accounts for connectivity and minimum degree at least two. And

$$\begin{aligned}A_i &= \sum_{l\ge 2}\sum_{a+b=l}\binom{M_i}{a}\binom{N_i}{b}p^l(1-p)^{n-1-l}\frac{a(a-1)}{l(l-1)}\\ &= \sum_{l\ge 2}p^l(1-p)^{n-1-l}\frac{M_i(M_i-1)}{l(l-1)}\sum_{a+b=l}\binom{M_i-2}{a-2}\binom{N_i}{b}\\ &= \sum_{l\ge 2}p^l(1-p)^{n-1-l}\frac{M_i(M_i-1)}{l(l-1)}\binom{n-3}{l-2}\\ &= \frac{M_i(M_i-1)p^2}{(n-1)(n-2)p^2}\sum_{l\ge 0}\binom{n-1}{l}p^l(1-p)^{n-1-l}\\ &= \frac{M_i(M_i-1)}{(n-1)(n-2)}.\end{aligned} \tag{8.6}$$

Explanation of (8.5). Fix disjoint sets K, L of size k and ϵk, respectively. Then for $v \in K$, A_1 will be the probability that both its chosen neighbors are in $K \cup L$, and for $v \notin K \cup L$, A_2 will be the probability that both its chosen neighbors are not in K. Now for $v \in K$, the number of $G_{n,p}$ neighbors in $K \cup L$ will be $a = Bin(M_1, p)$ and the number of $G_{n,p}$ neighbors not in $K \cup L$ will be $b = Bin(N_1, p)$. The probability of this is $\binom{M_1}{a}\binom{N_1}{b}p^l(1-p)^{n-1-l}$, where $l = a + b$. The probability that both of v's choices are in $K \cup L$ is $\binom{a}{2}/\binom{l}{2}$. When $v \notin K \cup L$, the number of $G_{n,p}$ neighbors not in K will be $a = Bin(M_2, p)$ and the number of $G_{n,p}$ neighbors in K will be $b = Bin(N_2, p)$. The probability of this is $\binom{M_2}{a}\binom{N_2}{b}p^l(1-p)^{n-1-l}$, where $l = a + b$. The probability that both of v's choices are not in K is $\binom{a}{2}/\binom{l}{2}$. Finally note that the described events are independent for each possible v.

It follows from (8.6) that

$$A_1 \leq \left(\frac{k+\epsilon k}{n}\right)^2 \text{ and } A_2 \leq \left(\frac{n-k}{n}\right)^2.$$

Comparing (8.4) and (8.5) we see that w.h.p. H has constant vertex expansion. The lower bound proof of Theorem 1.5 rests on the fact that H' has constant vertex expansion, and so this proof can be repeated here. □

9. Discussion. The problem of scalable routing in the presence of failures motivated our constructions in this paper. The use of trees is particularly natural for routing. Our results suggest using a constant number of trees in total for routing, as opposed to the norm of one or more trees per destination. Further, the manner in which the trees are obtained is simple to implement and can lead to faster recovery since (a) paths exist after several failures and (b) fewer trees need to be recomputed in any case.

One aspect of splicers that we have not explored is the stretch of the metric induced by them. For the case of the complete graph, it is not hard to see that the diameter is $O(\log n)$ and hence so is the expected stretch for a pair of random vertices. This continues to hold for $G_{n,p}$, in fact giving better bounds for small p (expected stretch of $O(\log \log n)$ for $p = \text{ poly}(\log n)/n$). It remains to study the stretch of splicers for arbitrary graphs or bounded-degree graphs. This seems to be an interesting question since on the complete graph, the expected stretch on one tree is $\Theta(\sqrt{n})$ while that of two trees is $O(\log n)$.

One future direction of research is to understand the trade-off between fault-tolerance and stretch achievable by splicers (not necessarily union of uniformly random spanning trees, but more carefully chosen spanning trees).

REFERENCES

[1] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup, *Compact name-independent routing with minimum stretch*, in Proceedings of the 16th Annual ACM Symposium on Parallelism in Algorithms and Architectures, New York, NY, ACM, 2004, pp. 20–24.

[2] D. Aldous, *The random walk construction of uniform spanning trees and uniform labelled trees*, SIAM J. Discrete Math., 3 (1990), pp. 450–465.

[3] N. Alon and M. R. Capalbo, *Finding disjoint paths in expanders deterministically and online*, in FOCS, 2007, pp. 518–524.

[4] N. Alon and J. H. Spencer, *The Probabilistic Method*, 2nd ed., Wiley-Intersci. Ser. Discrete Math. Optim., John Wiley & Sons, New York, 2000.

[5] J. Batson, D. A. Spielman, and N. Srivastava, *Twice-Ramanujan sparsifiers*, arXiv: 0808.0163v1, 2008.

[6] A. A. Benczúr and D. R. Karger, *Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time*, in Proceedings of STOC, 1996, pp. 47–55.
[7] B. Bollobás, *Random Graphs*, 2nd ed., Cambridge University Press, Cambridge, 2001.
[8] A. Z. Broder, *Generating random spanning trees*, in Proceedings of FOCS, 1989, pp. 442–447.
[9] D. P. Dubhashi and D. Ranjan, *Balls and bins: A study in negative dependence*, Random Structures Algorithms, 13 (1998), pp. 99–124.
[10] T. Feder and M. Mihail, *Balanced matroids*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, New York, NY, ACM, 1992, pp. 26–38.
[11] O. Goldreich, *Randomized Methods in Computation, Lecture* 2, http://www.wisdom.weizmann.ac.il/~oded/rnd.html (2001).
[12] M. Jerrum, A. Sinclair, and E. Vigoda, *A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries*, J. ACM, 51 (2004), pp. 671–697.
[13] F. T. Leighton and S. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, J. ACM, 46 (1999), pp. 787–832.
[14] L. Lovász, *Random walks on graphs: A survey*, in Combinatorics, Paul Erdős is Eighty, Vol. 2, D. Miklós, V. T. Sós, and T. Szőnyi, eds., János Bolyai Mathematical Society, Budapest, 1996, pp. 353–398.
[15] R. Lyons and Y. Peres, *Probability on Trees and Networks*, preprint http://mypage.iu.edu/~rdlyons/prbtree/prbtree.html (2005).
[16] M. Mihail, C. Papadimitriou, and A. Saberi, *On certain connectivity properties of the internet topology*, J. Comput. System Sci., 72 (2006), pp. 239–251.
[17] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala, *Path splicing*, in Proceedings of ACM SIGCOMM, Seattle, WA, 2008.
[18] M. Motiwala, N. Feamster, and S. Vempala, *Path splicing: Reliable connectivity with rapid recovery*, in Proceedings of the 6th ACM SIGCOMM HotNets Workshop, Atlanta, GA, 2007.
[19] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, 1995.
[20] R. Pemantle, *Toward a theory of negative dependence*, J. Math. Phys., 41 (2000), pp. 1371–1390.
[21] J. Propp and D. Wilson, *How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph*, J. Algorithms, 27 (1998), pp. 170–217.
[22] P. Raghavan and C. D. Thompson, *Randomized rounding: A technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7 (1987), pp. 365–374.
[23] S. Shenker, *We dream of geni: Exploring radical network designs*, in Proceedings of FCRC, http://lazowska.cs.washington.edu/fcrc/Shenker.FCRC.pdf (2007).
[24] D. Spielman and N. Srivastava, *Graph sparsification by effective resistances*, in Proceedings of STOC, 2008.