Disjoint Paths in Expander Graphs via Random Walks: a Short Survey

Alan M. Frieze*
Department of Mathematical Sciences
Carnegie-Mellon University
Pittsburgh
PA 15213
USA

Abstract

There has been a significant amount of research lately on solving the edge disjoint path and related problems on expander graphs. We review the random walk approach of Broder, Frieze and Upfal.

1 Introduction

The basic problem discussed in this paper can be described as follows: we are given a graph $G = (V, E)$ and a set of K disjoint pairs of vertices in V. If possible, find edge disjoint paths P_i that join a_i to b_i for $i = 1, 2, \ldots, K$. We call this the Edge Disjoint Paths problem. We also say that G is K-*routable* if such paths exist for *any* set of K pairs. For arbitrary graphs, deciding whether such paths exist is in $\mathcal{P}$ for fixed K – Robertson and Seymour [16], but is $\mathcal{NP}$-complete if K is part of the input, being one of Karp's original problems. This negative result can be circumvented for certain classes of graphs, see Frank [7]. In this paper we will focus on *expander graphs*. There have been essentially two bases for approaches to this problem in this context: (i) random walks and (ii) multicommodity flows. Our aim here is to provide a summary of the results known to us at present together with an outline of some of their proofs. We emphasise the random walk approach, see [11, 12, 13] for more detail on the multicommodity flow approach.

Expander Graphs For certain bounded degree expander graphs, Peleg and Upfal [15] showed that if G is a sufficiently strong expander then G is n^ϵ-routable for some small constant $\epsilon \ll 1/3$ that depends only on the expansion properties of the input graph. Furthermore there is a polynomial time algorithm for constructing such paths.

This result has now been substantially improved and there is only a small factor (essentially $\log n$) between upper and lower bounds for maximum routability.

*Supported in part by NSF grant CCR9530974. E-mail: **alan@random.math.cmu.edu**.

Using random walks, Broder, Frieze and Upfal [2] improved the result of [15] to obtain the same result for $K = n/(\log n)^{\kappa}$ where κ depends only on the expansion properties of the graph. More recently, they [3] improved this by replacing κ by $2+\epsilon$ for any positive constant $\epsilon > 0$, at the expense of requesting greater expansion properties of G. More recently still, Leighton, Rao and Srinivasan [13], using the rival multi-commodity flow technology have improved on this by showing that the ϵ can be replaced by $o(1)$. In Section 4 we will show how the random walks approach can be improved to give the same result, Theorem 3. It is rather interesting that these, in many ways quite different, approaches seem to yield roughly the same results. We note that both approaches yield a non-constructive proof [3], [12] (via the local lemma) that in a sufficiently strong expander $K = \Omega(n/(\log n)^2)$ is achievable.

Random Graphs Random graphs are well known to be excellent expanders and so it is perhaps not surprising that they very highly "routable". Broder, Frieze, Suen and Upfal [4] and Frieze and Zhao [9] (see Theorems 7,8) show that they are K-routable where K is within a constant factor of a simple lower bound, something that has not yet been achieved for arbitrary expander graphs.

Low Congestion Path Sets One way of generalising the problem is to bound the number of paths that use any one edge, the *edge congestion*, by some value g in place of one. Bounds on the number of routable pairs in this case are given in Theorem 5.

Dynamic problem In the dynamic version of the problem each vertex receives an infinite stream of requests for paths starting at that vertex. The times between requests are random and paths are are only required for a certain time (until the communication terminates) and then the path is deleted. Again each edge in the network should not be used by more than g paths at once.

The random walk approach gives a simple and fully distributed solution for this problem. In [3] (see Theorem 6) we show that if the injection to the network and the duration of connections are both controlled by Poisson processes then there is an algorithm which achieves a steady state utilization of the network which is similar to the utilization achieved in the *static* case situation, Theorem 5.

Approximation Algorithm So far we have only considered the case where all requests for paths have to be filled. If this is not possible then one might be satisfied with filling as many requests as possible. Kleinberg and Rubinfeld [10] (see Theorem 10) prove that a certain greedy strategy provides has a worst-case performance ratio of order $1/(\log n \log\log n)$.

Vertex Disjoint Paths Finally, there is the problem of finding vertex disjoint paths between a given set of pairs of vertices. In the worst-case one cannot do better than the minimum degree of the graph. The interest therefore must be on graphs with degrees which grow with the size of the graph. In this context random graphs [5] have optimal routing properties, to within a constant factor.

The structure of the paper is now as follows: Section 3 discusses the problem of splitting an expander, a basic requirement for finding edge disjoint paths. Section 4 details the aforementioned results on expander graphs and outlines some of the proofs for the random walk approach. Section 5 details the results on random graphs and outlines the corresponding proofs. Section 6 describes the result of Kleinberg and Rubinfeld. A final section provides some open problems.

2 Preliminaries

There are various ways to define expander graphs; here we define them in terms of edge expansion (a weaker property than vertex expansion).

For a set of vertices $S \subset V$ let $\mathrm{out}(S)$ be the number of edges with one end-point in S and one end-point in $V \setminus S$, that is

$$\mathrm{out}(S) = \left| \Big\{ \{u,v\} \mid \{u,v\} \in E, u \in S, v \notin S \Big\} \right|.$$

Similarly,

$$\mathrm{in}(S) = \left| \Big\{ \{u,v\} \mid \{u,v\} \in E, u, v \in S \Big\} \right|.$$

Definition 1 *A graph $G = (V, E)$ is a β-expander, if for every set $S \subset V$, $|S| \leq |V|/2$, we have $\mathrm{out}(S) \geq \beta|S|$.*

For certain results we need expanders that have the property that the expansion of small sets is not too small. The form of definition given below is taken from [3].

Definition 2 *An r-regular graph $G = (V, E)$ is called an (α, β, γ)-expander if for every set $S \subset V$*

$$\mathrm{out}(S) \geq \begin{cases} (1-\alpha)r|S| & \text{if } |S| \leq \gamma|V| \\ \beta|S| & \text{if } \gamma|V| < |S| \leq |V|/2 \end{cases}$$

In particular random regular graphs and the (explicitly constructible) Ramanujan graphs of of Lubotsky, Phillips and Sarnak [14] are (α, β, γ)-expanders with $\alpha = O(\gamma + \frac{1}{r^{1/2}})$ and β close to $r/4$.

A *random walk* on an undirected graph $G = (V, E)$ is a Markov chain $\{X_t\} \subseteq V$ associated with a particle that moves from vertex to vertex according to the following rule: the probability of a transition from vertex i, of degree d_i, to vertex j is $1/d_i$ if $\{i, j\} \in E$, and 0 otherwise. (In case of a bi-partite graph we need to assume that we do nothing with probability 1/2 and move off with probability 1/2 only. This technicality is ignored for the remainder of the paper.) Its stationary distribution, denoted π, is given by $\pi(v) = d_v/(2|E|)$. Obviously, for regular graphs, the stationary distribution is uniform.

A trajectory W of length τ is a sequence of vertices $[w_0, w_1, \ldots, w_\tau]$ such that $\{w_t, w_{t+1}\} \in E$. The Markov chain $\{X_t\}$ induces a probability distribution on trajectories, namely the product of the probabilities of the transitions that define the trajectory.

Let P denote the transition probability matrix of the random walk on G, and let $P^{(t)}_{v,w}$ denote the probability that the walk is at w at step t given that it started at v. Let λ be the second largest eigenvalue of P. (All eigenvalues of P are real.) It is known that

$$|P^{(t)}_{v,w} - \pi(w)| \leq \lambda^t \sqrt{\pi(w)/\pi(v)}. \tag{1}$$

In particular, for regular graphs

$$P_{v,w}^{(t)} = \frac{1}{n} + O(\lambda^t). \tag{2}$$

To ensure rapid convergence we need $\lambda \leq 1 - \epsilon$ for some *constant* $\epsilon > 0$. This holds for all expanders (Alon [1]). In particular if G is a β-expander with maximum degree Δ respectively then Jerrum and Sinclair [17] show that

$$\lambda \leq 1 - \frac{1}{2}\left(\frac{\beta}{\Delta}\right)^2 \tag{3}$$

It is often useful to consider the *separation* s of the distribution $P_{v,\cdot}^{(t)}$ from the limit distribution π given by

$$s(t) = \max_{v,w} \frac{\pi(w) - P_{v,w}^{(t)}}{\pi(w)}. \tag{4}$$

Then we can write

$$P_{v,\cdot}^{(t)} = (1 - s(t))\pi + s(t)\sigma$$

where σ is a probability distribution. We can then imagine that the distribution $P_{v,\cdot}^{(t)}$ is producing by choosing either σ with probability $s(t)$ or π with probability $1 - s(t)$. Hence if $\mathcal{E}$ is an event that depends only on the state of the Markov chain we have

$$(1 - s(t))\mathbf{Pr}(\mathcal{E} \text{ under } \pi) + s(t) \geq \mathbf{Pr}(\mathcal{E} \text{ under } P_{v,\cdot}^{(t)}) \geq (1 - s(t))\mathbf{Pr}(\mathcal{E} \text{ under } \pi). \tag{5}$$

We use this in the following scenario:

Experiment A: Choose $u_1 \in V$ with distribution π and do a random walk W_1 of length τ from u_1. Let v_1 be the terminal vertex of W_1.

Experiment B: Choose u_2 and v_2 independently from V with distribution π and do a random walk of length τ from u_2 to v_2.

We claim that for any event $\mathcal{E}$ depending on walks of length τ,

$$|\mathbf{Pr}((u_1, v_1, W_1) \in \mathcal{E}) - \mathbf{Pr}((u_2, v_2, W_2) \in \mathcal{E})| \leq s(\tau). \tag{6}$$

This follows from the stronger claim that for any $u \in V$ and any event $\mathcal{E}$ depending on walks of length τ

$$|\mathbf{Pr}((u_1, v_1, W_1) \in \mathcal{E} \mid u_1 = u) - \mathbf{Pr}((u_2, v_2, W_2) \in \mathcal{E} \mid u_2 = u)| \leq s(\tau),$$

which follows from (5).

The notation $B(m, p)$ stands for the binomial random variable with parameters $m =$ number of trials, and $p =$ probability of success.

3 Splitting an Expander

Most of the algorithms we describe work in phases. Each phase generates paths and it is important that the sets of paths produced in each phase remain edge disjoint. One way of ensuring this is to insist that different phases work on different expander graphs. If the input consists of a single expander then we need a procedure for partitioning E into p sets, say $E_1, E_2, \ldots, E_p$, where the graphs $G_i = (V, E_i)$ are themselves expanders.

A natural way of trying to split G into expander graphs is to randomly partition E into p sets. The problem with this is that in a bounded degree expander this will almost surely lead to subgraphs with isolated vertices. We must find a partition which provides a high minimum degree in both graphs. The solution in [2, 3] is

Algorithm Partition(G, p)

1. Orient the edges of G so that $|outdegree(v) - indegree(v)| \leq 1$ for all $v \in V$.
2. For each $v \in V$ randomly partition the edges directed out of v into p sets $X_{v,1}, \ldots, X_{v,p}$ each of size $\lfloor r/2p \rfloor$ or $\lceil r/2p \rceil$. Let $E_i = \bigcup_{v \in V} X_{v,i}$, for $1 \leq i \leq p$.

Define $H(\gamma) = ((1-\gamma)^{1-\gamma}\gamma^{\gamma})^{-1}$ and $\psi(\epsilon) = (1-\epsilon)\ln(1-\epsilon) + \epsilon$.

Theorem 1 *Let $G = (V, E)$ be an r-regular n-vertex graph that is an (α, β, γ)-expander. If $\epsilon \in (0,1)$ and $p \leq r/2$ are such that $\beta > p(\gamma\psi(\epsilon))^{-1}\ln(H(\gamma))$, then Partition splits the edge-set of G into p subgraphs. With probability at least* [1] *$1 - \exp(-n(\beta\gamma\psi(\epsilon)/p - \ln(H(\gamma)) - o(1)))$, all the p subgraphs span V and have edge-expansion at least*

- *$\lfloor r/(2p) \rfloor - \alpha r$ for sets of size at most γn.*
- *$(1-\epsilon)\beta/p$ for sets of size betweem γn and $n/2$.*

In particular each G_i is a ζ-expander where

$$\zeta = \min\{\lfloor r/(2p) \rfloor - \alpha r, (1-\epsilon)\beta/p\}. \tag{7}$$

This does not seem to be the best way to proceed, but it is the best we know constructively. Frieze and Molloy [8] have a stronger result which is close to optimal, but at present it is non-constructive. Let the edge-expansion $\eta(G)$ of G be defined by

$$\eta(G) = \min_{\substack{S \subseteq V \\ |S| \leq n/2}} \frac{\mathrm{out}(S)}{|S|}.$$

Theorem 2 *Let $p \geq 2$ be a positive integer and let $\epsilon > 0$ be a small positive real number. Suppose that*

$$\begin{aligned} \frac{r}{\log r} &\geq 7p\epsilon^{-2} \\ \eta(G) &\geq 4\epsilon^{-2} p \log r. \end{aligned}$$

Then there exists a partition $E = E_1 \cup E_2 \cup \cdots \cup E_p$ such that for $1 \leq i \leq p$

[1] The $o(1)$ term tends to 0 as $n \to \infty$.

(a)

$$\eta(G_i) \geq \frac{(1-\epsilon)\eta(G)}{p}.$$

(b)

$$\frac{(1-\epsilon)r}{p} \leq \delta(G_i) \leq \Delta(G_i) \leq \frac{(1+\epsilon)r}{p}.$$

4 Finding paths in Expander graphs

4.1 Edge Disjoint paths

We will first concentrate on showing how using random walks we can achieve the same bound on the number of routable pairs as given in [13].

Fix integer $s \geq 1$ and let $\log^{(s)}$ denote the natural logarithm iterated s times e.g. $\log^{(2)} n = \log\log n$.

Let $m_1 = \log^{(s)} n$ and $m_{i+1} = \lceil \frac{1}{4}\left(\frac{10}{3e}\right)^{m_i/10} \rceil$ for $i \geq 1$. Then let $K_i = \lfloor 2crn/(m_i(\log n)^2) \rfloor$ for $i \geq 1$ and $\sigma \leq s+2$ be the largest i such that $K_i > 0$. Here $c = O(\zeta^2/(r^2 s))$ is a positive constant – ζ as in (7).

Theorem 3 *Suppose G is an r-regular n-vertex graph that is an (α,β,γ)-expander. Suppose that $\zeta > 1$ above when $p = 5\sigma$. Then G is $crn/((\log n)^2 \log^{(s)} n)$ -routable.*

Proof We first split G into $p = 5\sigma$ expander graphs $G_1, G_2, \ldots, G_p$ using algorithm Partition. Note that the minimum degree in each G_i is at least $r/(2p)$ and maximum degree is at most $r/2$.

Let $H_i = G_{5(i-1)+1} \cup \cdots \cup G_{5i}$ for $1 \leq i \leq \sigma$. The algorithm runs in phases. Phase i is left to deal with at most $\lfloor K_i \rfloor$ pairs left over from Phases 1 to $i-1$, assuming these phases have all succeeded. Thus Phase σ should finish the job. We run Phase i on graph H_i and this keeps the paths edge disjoint. Denote the set of source-sink pairs for Phase i by $S_{A,i} = \{a_{1,i}, \ldots, a_{K_i,i}\}$ and $S_{B,i} = \{b_{1,i} \ldots, b_{K_i,i}\}$. Phase i is divided into 4 subphases.

Subphase i.a: The aim here is to choose w_j, W_j, $1 \leq j \leq 2K_i$ such that (i) $w_j \in W_j$, (ii) $|W_j| = m_i + 1$, (iii) the sets $W_j, 1 \leq j \leq 2K_i$ are pairwise disjoint and (iv) W_j induces a connected subgraph of $\Gamma_i = G_{5(i-1)+2}$.

As in [11] we can partition an arbitrary spanning tree T of Γ_i. Since T has maximum degree at most r we can find $2K_i$ vertex disjoint subtrees T_j, $1 \leq j \leq 2K_i$ of T, each containing between m_i+1 and $(r-1)m_i+2$ vertices. We can find T_1 as follows: choose an arbitrary root ρ and let $Q_1, Q_2, \ldots, Q_\sigma$ be the subtrees of ρ. If there exists l such that Q_l has between m_i+1 and $(r-1)m_i+2$ vertices then we take $T_1 = Q_l$. Otherwise we can search for T_1 in any Q_ℓ with more than $(r-1)m_i+2$ vertices. Since $T \setminus T_1$ is connected, we can choose all of the T_j's in this way. Finally, W_j is the vertex set of an arbitrary m_i+1 vertex subtree of T_j and w_j is an arbitrary member of W_j for $j = 1, 2, \ldots, 2K_i$.

Subphase i.b: Using a network flow algorithm in $G_{5(i-1)+1}$ connect in an arbitrary manner the vertices of $S_{A,i} \cup S_{B,i}$ to $W_i = \{w_1, \ldots, w_{2K_i}\}$ by $2K_i$ edge disjoint paths as follows:

- Assume that every edge in $G_{5(i-1)+1}$ has a capacity equal to 1.
- View each vertex in S_i as a source with capacity 1 and similarly every vertex in W_i as a sink with capacity equal 1.

The expansion properties of $G_{4(i-1)+1}$ ensure that such flows always exist.

Let $\tilde{a}_{k,i}$ (resp. $\tilde{b}_{k,i}$) denote the vertex in W_i that was connected to the original endpoint $a_{k,i}$ (resp. $b_{k,i}$). Our problem is now to find edge disjoint paths joining $\tilde{a}_{k,i}$ to $\tilde{b}_{k,i}$ for $1 \leq k \leq K_i$.

Subphase i.c: If w_t has been renamed as $\tilde{a}_{k,i}$ (resp. $\tilde{b}_{k,i}$) then rename the elements of W_t as $\tilde{a}_{k,\ell,i}$, (resp. $\tilde{b}_{k,\ell,i}$,) $1 \leq \ell \leq m_i$. Choose $\xi_j, 1 \leq j \leq m_iK_i$ and $\eta_j, 1 \leq j \leq m_iK_i$ independently at random from the steady state distribution π_i of a random walk on G_{5i}. Using a network flow algorithm as in Subphase i.b, connect $\{\tilde{a}_{k,\ell,i} : 1 \leq k \leq K_i, 1 \leq \ell \leq m_i\}$ to $\{\xi_j : 1 \leq j \leq m_iK_i\}$ by edge disjoint paths in G_{5i-2}. Similarly, connect $\{\tilde{b}_{k,\ell,i} : 1 \leq k \leq K_i, 1 \leq \ell \leq m_i\}$ to $\{\eta_j : 1 \leq j \leq m_iK_i\}$ by edge disjoint paths in G_{5i-1}. Rename the other endpoint of the path starting at $\tilde{a}_{k,\ell,i}$ (resp. $\tilde{b}_{k,\ell,i}$) as $\hat{a}_{k,\ell,i}$ (resp. $\hat{b}_{k,\ell,i}$).

Once again the expansion properties of G_{5i-2}, G_{5i-1} ensure that flows exist.

Subphase i.d: Choose $\hat{x}_{k,\ell,i}, 1 \leq k \leq K_i, 1 \leq \ell \leq m_i$ independently at random from the steady state distribution π_i of a random walk on G_{5i}. Let $W'_{k,\ell,i}$ (resp. $W''_{k,\ell,i}$ be a random walk of length $\theta \log n$ from $\hat{a}_{k,\ell,i}$ (resp. $\hat{b}_{k,\ell,i}$) to $\hat{x}_{k,\ell,i}$. Here $\theta = r^2/(2\zeta^2)$) is chosen so that the separation (4) between π_i and the distribution of the terminal vertex of the walk is $O(n^{-3})$. ((3) gives $\lambda_i \leq 1 - 2\zeta^2/r^2$ where λ_i is the second largest eigenvalue of a random walk on G_{5i}.) The use of this intermediate vertex $\hat{x}_{k,\ell,i}$ helps to break some conditioning caused by the pairing up of the flow algorithm.

Let $B'_{k,i}$ (resp. $B''_{k,i}$) denote the *bundle* of walks $W'_{k,\ell,i}, 1 \leq \ell \leq m_i$ (resp. $W''_{k,\ell,i}, 1 \leq \ell \leq m_i$). Following [13] we say that $W'_{k,\ell,i}$ is *bad* if there exists $k' \neq k$ such that $W'_{k,\ell,i}$ shares an edge with a walk in a bundle $B'_{k',i}$ or $B''_{k',i}$. Each walk starts at an independently chosen vertex and moves to an independently chosen destination. The steady state of a random walk is uniform on edges and so at each stage of a walk, each edge is equally likely to be crossed. Thus

$$\mathbf{Pr}(W'_{k,\ell,i} \text{ is bad}) \leq \frac{10 m_i K_i \theta^2 (\log n)^2 \sigma}{rn} \leq 1/10$$

for sufficiently small c.

We say that index k is bad if either $B'_{k,i}$ or $B''_{k,i}$ contain more than $m_i/3$ bad walks. If index k is not bad then we can find a walk from $\hat{a}_{k,\ell,i}$ to $\hat{b}_{k,\ell,i}$ through $\hat{x}_{k,\ell,i}$ for some ℓ which is edge disjoint from all other walks. This gives a walk

$$a_{k,i} \to \tilde{a}_{k,i} \to \tilde{a}_{k,\ell,i} \to \hat{a}_{k,\ell,i} \to \hat{x}_{k,\ell,i} \to \hat{b}_{k,\ell,i} \to \tilde{b}_{k,\ell,i} \to \tilde{b}_{k,i} \to b_{k,i},$$

which is edge-disjoint from all other such walks.

The probability that index k is bad is at most

$$2\mathbf{Pr}(B(m_i, .1) \geq m_i/3) \leq 2(3e/10)^{m_i/10}.$$

So with probability at least 1/2 the number of bad indices is no more than $\lfloor 4K_i(3e/10)^{m_i/10}\rfloor \leq K_{i+1}$. By repetition we can ensure that Phase i succeeds **whp**. The theorem follows. $\square$

4.1.1 Existence Result

In this section we describe the use of the Lovász Local Lemma [6] to prove the *existence* of a large number of edge disjoint paths in any r-regular (α, β, γ)-expander, [3]. At the present time we do not see how to make the argument constructive.

Theorem 4 *Given a bounded degree (α, β, γ)-expander graph there exists a parameter c that depends on α, β, γ, but not on n, such that any set of less than $cn/(\log n)^2$ disjoint pairs of vertices can be connected by edge disjoint paths.*

The proof starts by splitting G into 2 $\beta' > 1$ expanders and using the first to route $a_1, \ldots, b_K$ to randomly chosen $\tilde{a}_1, \ldots, \tilde{b}_K$ via edge disjoint paths found through a flow algorithm as in say, Subphase i.b of the algorithm of the previous section.

Then, for $1 \leq i \leq K$, $\tilde{a}_i$ is joined to $\tilde{b}_i$ via an $O(\log n)$ random walk W_i through a randomly chosen intermediate vertex x_i. We use the local lemma to show that $W_1, \ldots, W_K$ are edge disjoint with positive probability. Ignoring several technical problems we consider *bad* event $\mathcal{E}_{i,j} = \{W_i \cap W_j \neq \emptyset\}$ and argue that $\mathcal{E}_{i,j}$ depends only on the $< 2K$ events of the form $\mathcal{E}_{i',j}$ or $\mathcal{E}_{i,j'}$. Since $\mathbf{Pr}(\mathcal{E}_{i,j}) = O((\log n)^2/n)$ we can follow through if $K(\log n)^2/n \ll 1$. This gives the theorem.

4.2 Low Congestion Paths

We discuss the following result from [3].

Theorem 5 *There is an explicit polynomial time algorithm that can connect any set of $K = \alpha(n)n/\log n$ pairs of vertices on a bounded degree expander so that no edge is used by more than g paths where*

$$g = \begin{cases} O\left(s + \left\lceil \dfrac{\log\log n}{\log(1/\hat{\alpha})} \right\rceil\right), & \text{for } \alpha < 1/2; \\ O(s + \alpha + \log\log n), & \text{for } \alpha \geq 1/2, \end{cases}$$

$\hat{\alpha} = \min(\alpha, 1/\log\log n)$, and s is the maximal multiplicity of a vertex in the set of pairs.

See [11] for similar results proved via multi-commodity flows.

The algorithm uses the same flow/random walk paradigm that we have already seen twice above. $a_1, \ldots, b_K$ are joined to randomly chosen $\tilde{a}_1, \ldots, \tilde{b}_K$ via edge disjoint paths found through a flow algorithm. Then, for $1 \leq i \leq K$, $\tilde{a}_i$ is joined to $\tilde{b}_i$ via an $O(\log n)$ random walk W_i through a randomly chosen intermediate vertex x_i. The number of paths which use an edge is bounded by the sum of two binomials. We then see that for a sufficiently large $\kappa > 0$ **whp** there are fewer than $n/(\log n)^\kappa$ edges which have congestion greater than g. We delete all of the paths through such edges and re-join the corresponding pairs via edge disjoint paths, using the algorithm of [2]. □

Leighton, Rao and Srinvasan [12] generalise Theorem 4 by showing that for any $B \geq 1$, given enough expansion one can join $cn/(\log n)^{1+1/B}$ pairs with congestion at most B.

4.3 Dynamic Allocation of Paths

Broder, frieze and Upfal [3] discuus a stochastic model for studying a dynamic version of the circuit switching problem. In their model new requests for establishing paths arrive continuously at nodes according to a discrete Poisson process. Requests wait in the processor's queue until the requested path is established. The duration of a path is exponentially distributed.

Their model is characterized by three parameters:

- P_1 is an upper bound on the probability that a new request arrives at a given node at a given step.
- P_2 is the probability that a given existing path is terminated in a given step. A path *lives* from the time it is established until it is terminated.
- g is the maximum congestion allowed on any edge.
- The destinations of path requests are chosen uniformly at random among all the graph vertices

They study a simple and fully distributed algorithm for this problem. In the algorithm each processor at each step becomes active with a probability $P_1' > P_1$. An inactive processor does not try to establish a path even if there are requests in its queue. The algorithm can be succinctly described: Assume that a is active at step t, and the first request in a's queue is for b. Processor a tries to establish a path to b by choosing a random trajectory of length $\tau = c_0 \log n$ connecting a to b. If the path does not use any edge with congestion greater than $g - 1$, the path is established, otherwise the request stays in the queue.

Theorem 6 *Let*

$$\Phi = \min\left\{\frac{1}{\log(grn)}, \frac{rg}{\tau^{(g+1)/g}}\right\}.$$

There exists a constant γ such that if $P_1 \leq \gamma\Phi P_2$, then the system is stable and the expected wait of a request in the queue is $O(1/P_1)$.

Before outlining the proof let us see the consequence of this theorem. Let $\mathbf{E}(N) = nP_1$ be the expected number of new requests that arrive at the system at a given step, and let $\mathbf{E}(D) = 1/P_2$ be the expected duration of a connection. For the system to be stable, the expected number of simultaneously active paths in the steady state must be at least $\mathbf{E}(N)\mathbf{E}(D) = nP_1/P_2$. Plugging $g = \log\log n/\log\omega$ for some ω in the range $[1, \log n]$ in the definition of Φ we get

$$\Phi = \Omega\left(\frac{1}{\omega \log n}\right).$$

Thus the theorem above implies that for such a congestion g, the system remains stable even if we choose P_1 and P_2 such that

$$\mathbf{E}(N)\mathbf{E}(D) = n\frac{P_1}{P_2} = \gamma n \Phi = \Omega\left(\frac{n}{w \log n}\right),$$

in which case the dynamic algorithm utilizes the edges of the network almost as efficiently as the static algorithm, Theorem 5 (there seems to be an efficiency gap of maximum order $\log\log\log n$ for $\omega \leq \log\log n$).

In the proof of the theorem, time is partitioned into intervals of length $T \leq 1/(4P_1)$. Let H_t denote the history of the system during the first t time intervals. Define the event

$$\mathcal{E}(v,t) = \left\{\begin{array}{l}\text{If the queue of processor } v \text{ was not empty at the beginning of interval} \\ t \text{ then } v \text{ served at least one request during interval } t\end{array}\right\}$$

The goal is to show that for all v and t,

$$\mathbf{Pr}(\mathcal{E}(v,t) \mid H_{t-2}) > \frac{1}{2}. \tag{8}$$

Given this we conclude that in any segment of $2T$ steps processor v is serving at least one request with probability at least $1/2$. The number of new arrivals in this time interval has a Binomial distribution with expectation at most $2TP_1 < 1/2$. Thus, under these conditions the queue is dominated by an $M/M/1$ queue with expected inter-arrival distribution greater than $4T$, and expected service time smaller than $4T$. The queue is stable, and the expected wait in the queue is $O(1/T) = O(1/P_1)$.

To prove (8) we argue that with sufficiently high probability, (i) v becomes active at least once during an interval, (ii) there are no very old paths in the network, (iii) there are not too many paths in the network altogether and then (iv) we can argue that the first path that a processor v tries to establish is unlikely to use a fully loaded edge.

5 Random Graphs

We deal with two related models of a random graph. $G_{n,m}$ has vertex set $[n] = \{1, 2, \ldots, n\}$ and and exactly m edges, all sets of m edges having equal probability. The random graph $G_{r-\text{reg}}$ is uniformly randomly chosen from the set of r-regular graphs with vertex set $[n]$.

Let D be the median distance between pairs of vertices in graph $G_{n,m}$. Clearly it is not possible to connect more than $O(m/D)$ pairs of vertices by edge-disjoint paths, for all choices of pairs, since some choice would require more edges than all the edges available. In the case of bounded degree expanders, this absolute upper bound on k is $O(n/\log n)$. The results mentioned above use only a vanishing fraction of the set of edges of the graph, thus are far from reaching this upper bound. In contrast, Broder, Frieze, Suen and Upfal [4] and Frieze and Zhao [9] show that for $G_{n,m}$ and $G_{r-\text{reg}}$ the absolute upper bound is achievable within a constant factor, and present algorithms that construct the required paths in polynomial time.

Theorem 7 *Let $m = m(n)$ be such that $d = 2m/n \geq (1 + o(1)) \log n$. Then, as $n \longrightarrow \infty$, with probability $1 - o(1)$, the graph $G_{n,m}$ has the following property: there exist positive constants α and β such that for all sets of pairs of vertices $\{(a_i, b_i) \mid i = 1, ..., K\}$ satisfying:*

(i) $K \leq \lceil \alpha m \log d / \log n \rceil$,

(ii) *for each vertex v,* $|\{i : a_i = v\}| + |\{i : b_i = v\}| \leq \min\{d_G(v), \beta d\}$,

there exist edge-disjoint paths in G, joining a_i to b_i, for each $i = 1, 2, \ldots, K$. Furthermore, there is an $O(nm^2)$ time randomized algorithm for constructing these paths.

Theorem 8 *Let r be a sufficiently large constant. Then, as $n \longrightarrow \infty$, the graph G_{r-reg} has the following property* **whp***: there exist positive absolute constants α, β such that for all sets of pairs of vertices $\{(a_i, b_i) \mid i = 1, ..., K\}$ satisfying:*

(i) $K \leq \lceil \alpha r n / \log_r n \rceil$,

(ii) *for each vertex v,* $|\{i : a_i = v\}| + |\{i : b_i = v\}| \leq \beta r$,

there exist edge-disjoint paths in G_{r-reg}, joining a_i to b_i, for each $i = 1, 2, \ldots, K$. Furthermore, there is an $O(n^3)$ time randomized algorithm for constructing these paths.

These results are best possible up to constant factors. Consider for example Theorem 7. For (i) note that the distance between most pairs of vertices in $G_{n,m}$ is $\Omega(\log n / \log d)$, and thus with m edges we can connect at most $O(m \log d / \log n)$ pairs. For (ii) note that a vertex v can be the endpoint of at most $d_G(v)$ different paths. Furthermore suppose that $d \geq n^\gamma$ for some constant $\gamma > 0$ so that $K \geq \lceil \alpha \gamma n d / 2 \rceil$. Let $\epsilon = \alpha\gamma/3$, $A = [\epsilon n]$, and $B = [n] \setminus A$. Now with probability 1-o(1) there are less than $(1 + o(1))\epsilon(1 - \epsilon)nd$ edges between A and B in $G_{n,m}$. However almost all vertices of A have degree $(1 + o(1))d$ and if for these vertices we ask for $(1 - \epsilon/2)d$ edge-disjoint paths to vertices in B then the number of paths required is at most $(1 + o(1))\epsilon(1 - \epsilon/2)nd < K$, but, without further restrictions, this many paths would require at least $(1 - o(1))\epsilon(1 - \epsilon/2)nd > (1 + o(1))\epsilon(1 - \epsilon)nd$ edges between A and B which is more than what is available. This justifies an upper bound of $1 - \epsilon/2$ for β of Theorem 7. A similar argument justifies the bounds in Theorem 8.

First consider Theorem 7. The edge set E is first split randomly into 5 sets $E_1, E_2, \ldots, E_5$. The graphs $G_i = (V, E_i)$ will all be good expanders **whp**. As usual, G_1 is used to connect $a_1, \ldots, b_K$ to randomly chosen $\tilde{a}_1, \ldots, \tilde{b}_K$ using network flows. A random walk is then done in G_2 starting at each of these latter vertices and ending at $\hat{a}_1, \ldots, \hat{b}_K$. After each walk, the edges are deleted from G_2 which keeps the walks edge disjoint. The length τ of these walks is $O(\log n / \log d)$ but long enough so that the endpoints $\hat{a}_1, \ldots, \hat{b}_K$ are (essentially) independent of their start points. This handles any possible conditioning introduced by the pairing of $\tilde{a}_i$ with $\tilde{b}_i$. Finally, $\hat{a}_i$ is joined to $\hat{b}_i$ directly by a random walk of length τ in G_3. G_4 and G_5 and the algorithm of [2] are used to connect the few pairs not successfully joined by the above process.

The success of this algorithm rests on the fact that if not too many walks, $O(m \log d / \log n)$, are deleted then $O(m)$ edges will be deleted and we will be easily be able to ensure that the degrees of almost all vertices stay logarithmic in size. Thus the remaining graphs will be good expanders. The reader should notice that we only need a constant number of short random walks to connect each pair and this is why we are within a constant of optimal.

In Theorem 3 where degrees are bounded, we find that this argument breaks down because many (order n) vertices would become isolated through the deletion of the requested number of walks. The cure for this is to force the "action" to take place on a *core* of each subgraph (The k-core of a graph H is the largest subset of V which induces a subgraph of minimum degree at least k in H. It is unique and can be found by repeatedly removing vertices which have degree less than k.) This raises technical problems, such as what is to be done when one of the endpoints of a proposed walk drops out of the core. These problems are dealt with in [9].

The problem of finding vertex disjoint paths in random graphs is dealt with in Broder, Frieze, Suen and Upfal [5].

Theorem 9 *Suppose* $m = \frac{n}{2}(\log n + \omega)$ *where* $\omega(n) \to \infty$. *Then there exists* $\alpha, \beta > 0$ *such that* **whp** *for all* $A = \{a_1, a_2, \ldots, a_K\}, B = \{b_1, b_2, \ldots, b_K\} \subseteq [n]$ *satisfying*

(i) $A \cap B = \emptyset$

(ii) $|A| = |B| \leq \frac{\alpha n \ln d}{\log n}$

(iii) $|N(v) \cap (A \cup B)| \leq \beta |N(v)|$.

there are vertex disjoint paths P_i *from* a_i *to* b_i *for* $1 \leq i \leq K$. *Furthermore these paths can be constructed by a randomised algorithm in* $O(n^3)$ *time.*

This result is best possible up to constant factors.

6 Approximation Algorithm

Kleinberg and Rubinfeld [10] describe an on-line Bounded Degree (BGA) Approximation algorithm for the edge disjoint paths problem. BGA is defined by a parameter L as follows:

(i) Proceed through the terminal pairs in one pass.

(ii) When (a_i, b_i) is considered, check whether a_i and b_i can still be joined by a path of length at most L. If so, route (a_i, b_i) on such a path P_i. Delete P_i and iterate.

They prove the following:

Theorem 10 *Suppose* G *is an expander of maximum degree* Δ. *Then there exists* $c > 0$ *such that with* $L = c\Delta \log n$, *BGA is an* $O(\log n \log \log n)$*-approximation algorithm for the edge disjoint paths problem.*

7 Final Remarks

There has been a lot of progress since the first paper of Peleg and Upfal. The most interesting questions that remain to my mind are:

1. Can we take $K = \Omega(n/\log n)$, given sufficient expansion, in Theorem 3?
2. More modestly, can we remove the $\log^{(s)} n$ factor and make Theorem 4 constructive?
3. Can we achieve near optimal expander splitting as in Theorem 2, constructively?
4. Is there a constant factor approximation algorithm for the edge disjoint paths problem on expander graphs?
5. Can any of the above results be extended to digraphs?

References

[1] N.Alon, Eigenvalues and expanders, *Combinatorica* 6 (1986) 83-96.

[2] A.Z.Broder, A.M.Frieze and E.Upfal, Existence and construction of edge-disjoint paths on expander graphs, *SIAM Journal of Computing* 23 (1994) 976-989.

[3] A.Z.Broder, A.M.Frieze and E.Upfal, Existence and construction of edge low congestion paths on expander graphs, *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, (1997) 531-539.

[4] A.Z.Broder, A.M.Frieze, S.Suen and E.Upfal, Optimal construction of edge disjoint paths in random graphs, *Proceedings of 4th Annual Symposium on Discrete Algorithms*, (1994) 603-612.

[5] A.Z.Broder, A.M.Frieze, S.Suen and E.Upfal, An efficient algorithm for the vertex-disjoint paths problem in random graphs, *Proceedings of 6th Annual Symposium on Discrete Algorithms*, (1996) 261-268.

[6] P.Erdös and L.Lovász, Problems and results on 3-chromatic hypergraphs and some related questions, In A.Hajnal et al., editors, *Infinite and Finite Sets*, Volume 11 of *Colloq. Math. Soc. J. Bolyai* (1975)609-627.

[7] A.Frank, Packing paths, cuts and circuits – a survey, in *Paths, Flows and VLSI Layout*, B.Korte, L.Lovász, H.J.Prömel and A.Schrijver Eds., Springer-Verlag, 1990.

[8] A.M.Frieze and M.Molloy, Splitting expanders.

[9] A.M.Frieze and L.Zhao, Edge disjoint paths in random regular graphs.

[10] J.Kleinberg and R.Rubinfeld, Short paths in expander graphs, *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, (1996) 86-95.

[11] T.Leighton and S.Rao, Circuit switching: a multicommodity flow based approach, Proceedings of a Workshop on Randomized Parallel Computing 1996.

[12] T.Leighton, S.Rao and A.Srinivasan, Multi-commodity flow and circuit switching, *Proceedings of the Hawaii International Conference on System Sciences*, 1998.

[13] T.Leighton, S.Rao and A.Srinivasan, New algorithmic aspects of the local lemma with applications to partitioning and routing.

[14] A.Lubotsky, R.Phillips, and P.Sarnak, Ramanujan graphs, *Combinatorica* 8 (1988) 261-277.

[15] D.Peleg and E.Upfal, Constructing disjoint paths on expander graphs, *Combinatorica*, 9, (1989) 289-313.

[16] N.Robertson and P.D. Seymour, Graph minors-XIII: The disjoint paths problem.

[17] A.Sinclair and M.Jerrum, Approximate counting, uniform generation and rapidly mixing Markov chains, *Information and Computation* 82(1) (1989) 93-133.