Open access • Journal Article • DOI:10.1137/S0097539792232021

Existence and Construction of Edge-Disjoint Pathson Expander Graphs

Andrei Z. Broder, Alan Frieze, Eli Upfal

Published on: 01 Oct 1994 - SIAM Journal on Computing (Society for Industrial and Applied Mathematics)

Topics: Disjoint sets, Expander graph, Vertex (geometry) and Graph theory

Existence and Construction of Edge-Disjoint Paths on Expander Graphs

Andrei Z. Broder* Alan M. Frieze† Eli Upfal‡

Abstract

Given an expander graph $G = (V, E)$ and a set of q disjoint pairs of vertices in V, we are interested in finding for each pair (a_i, b_i), a path connecting a_i to b_i, such that the set of q paths so found is edge-disjoint. (For general graphs the related decision problem is NP-complete.)

We prove sufficient conditions for the existence of edge-disjoint paths connecting any set of $q \leq n/(\log n)^\kappa$ disjoint pairs of vertices on any n vertex bounded degree expander, where κ depends only on the expansion properties of the input graph, and not on n. Furthermore, we present a randomized $o(n^3)$ time algorithm, and a random $\mathcal{NC}$ algorithm for constructing these paths. (Previous existence proofs and construction algorithms allowed only up to n^ϵ pairs, for some $\epsilon \ll 1/3$, and strong expanders [19].)

In passing, we develop an algorithm for splitting a sufficiently strong expander into two edge-disjoint spanning expanders.

1 Introduction

Given an expander graph $G = (V, E)$ and a set of q disjoint pairs of vertices in V, we are interested in finding for each pair (a_i, b_i), a path connecting a_i

*DEC Systems Research Center, 130 Lytton Ave, Palo Alto, CA 94301.

†Department of Mathematics, Carnegie-Mellon University. A portion of this work was done while the author was visiting DEC SRC. Supported in part by NSF grants CCR8900112 and CCR9024935.

‡IBM Almaden Research Center, San Jose, CA 95120, and Department of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel.

to b_i, such that the set of q paths so found is edge-disjoint.

For arbitrary graphs, the related decision problem is in $\mathcal{P}$ for fixed q – Robertson and Seymour [21], but is $\mathcal{NP}$-complete if q is part of the input. However, this negative result can be circumvented for certain classes of graphs. For certain bounded degree expander graphs, Peleg and Upfal [19] have presented a polynomial time algorithm and a random $\mathcal{NC}$ algorithm for constructing up to n^ϵ disjoint paths, were $0 < \epsilon \ll 1/3$ is a constant that depends on the expansion properties of the input graph.

In this paper we describe a new algorithm for constructing edge-disjoint paths. Using it we can construct up to $n/(\ln n)^\kappa$ disjoint paths on bounded degree graphs with a sufficiently strong expansion property, where κ is a constant that depends only on the expansion property of the input graph. Our algorithm is based on two probabilistic tools: the rapid mixing properties of random walks on expanders, and the Lovász Local Lemma [10].

As in [19], the disjoint paths are constructed in two stages. In the first stage we choose a random set Q of $2q$ vertices that are at least $\kappa_1 \ln \ln n$ apart from each other. (The constant κ_1 will be defined later.) We connect the original endpoints to the vertices of Q in an arbitrary fashion via edge-disjoint paths, such that each Q-vertex is the endpoint of exactly one path. A simple flow argument proves constructively the existence of such edge-disjoint paths on any graph with edge expansion larger than one.

Let $\tilde{a}_i$ (resp. $\tilde{b}_i$) denote the vertex in Q that was connected to the original end-point a_i (resp. b_i) in the first stage. The core of the algorithm consists of constructing edge-disjoint paths connecting $\tilde{a}_i$ to $\tilde{b}_i$, for $i = 1, ..., q$.

In the second stage of the algorithm we choose for each pair $(\tilde{a}_i, \tilde{b}_i)$ a bundle of $(\ln n)^2$ random paths connecting them. The goal is to show via the Lovász Local Lemma that there is one choice out of each bundle of paths such that the set of chosen paths is edge-disjoint. However, the Local Lemma cannot be applied to sets of paths chosen uniformly at random, since the dependency graph that corresponds to such choices is too dense. The crux of the proof is that for any expander graph there is a simple pruning mechanism that reduces the dependency so that the Lemma can be applied.

To convert the existence proof to an explicit algorithm, we observe that the dependency graph constructed in this process is almost surely composed of sufficiently small components, such that the paths can be selected in polynomial time by exhaustive search.

Each of the two stages of the algorithm requires an expander graph. If

we apply both stages to the same expander we prove:

Theorem 1 *Given any n vertex, bounded degree, regular graph G with edge expansion greater than one, and given any set of $q \leq n/(\log n)^{\kappa}$ disjoint pairs of vertices in G, with high probability our algorithm finds in $o(n^3)$ steps a set of paths in G connecting the q pairs, such that each edge in G participates in no more than two paths. (The constant κ is exposed in the proof).*

While the above result is sufficient for most applications, it is still of theoretical interest to find edge-disjoint paths. To achieve this goal we develop a new algorithm for splitting a sufficiently strong expander into two edge-disjoint spanning expanders. We believe that this algorithm is of independent interest. The splitting algorithm requires a stronger, but still bounded degree expander. Splitting it, and applying each stage of our algorithm to a different set of edges we prove:

Theorem 2 *Given an n vertex graph with sufficiently strong edge expansion, and given any set of $q \leq n/(\log n)^{\kappa}$ disjoint pairs of vertices in G, with high probability, our algorithm finds in $o(n^3)$ steps a set of edge-disjoint paths in G connecting the q pairs. (The constant κ and the required edge expansion are defined later).*

The disjoint paths problem has numerous algorithmic applications. One that has received increased attention in recent years is in the context of communication networks for parallel and distributed computing. While packet routing is the communication protocol of choice for bounded size messages, it cannot always be used efficiently for high volume communication such as in multi-media applications and in two way communication. A more efficient way to transmit such information is through disjoint paths (virtual circuits) that are dedicated to one pair of processors for the duration of the communication. Our result gives yet more evidence of the usefulness of a communication network with strong expansion properties [24, 16, 20].

A preliminary version of this paper has appeared in [8].

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).

Let $G = (V, E)$ be a graph. For a set of vertices $S \subset V$ let $\text{out}(S)$ be the set of edges with one end-point in S and one end-point in $V \setminus S$, that is

$$\text{out}(S) = \Big\{\{u, v\} \mid \{u, v\} \in E, u \in S, v \notin S\Big\}.$$

Similarly

$$\text{in}(S) = \Big\{\{u, v\} \mid \{u, v\} \in E, u \in S, v \in S\Big\}.$$

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

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

$$\text{out}(S) \geq \begin{cases} (3r/4 + \alpha)|S| & \text{if } |S| \leq \gamma|V| \\ \beta|S| & \text{if } \gamma|V| < |S| \leq |V|/2 \end{cases}$$

Definition 3 *The t-neighborhood of a vertex u in G, is the set of all vertices that are at distance t or less from u in G.*

The notation $\text{dist}(u, v)$ refers to the distance from vertex v to vertex u. If U is a set of vertices, then $\text{dist}(U, v) = \min_{u \in U} \text{dist}(u, v)$. If P is a path (which we view as a set of edges), then $\text{dist}(P, v)$ is the minimum over all vertices u on P of $\text{dist}(u, v)$. These definitions generalize in the obvious manner when both arguments are paths, sets, etc.

A *random walk* on the undirected graph $G = (V, E)$ is a Markov chain $\{X_t\} \subseteq V$ associated to 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, (or $\pi(G)$) is given by $\pi_v = d_v/(2\,|E|)$. 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.

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

The notation $[m]$ stands for the set $\{1, 2, \ldots, m\}$ for any positive integer m.

3 The Sequential Algorithm

Below we present the algorithm for finding disjoint paths. If we relax the requirements so that edges can be used twice, then the input can be any β-expander, $\beta > 1$, and Phase 1 is entirely omitted. The algorithm involves certain constants $\kappa, \kappa_1, \kappa_2, \kappa_3$, and κ_4. The required relations between these constants are explicitly given in equations (13) – (15). At various points of the algorithm we stop if certain conditions fail to hold. The subsequent analysis shows that premature termination is unlikely.

Algorithm DisjPaths

Input: An r-regular graph $G = (V, E)$, with sufficiently strong expansion. A collection of q disjoint pairs of vertices $\{(a_1, b_1), \ldots, (a_q, b_q)\}$. (The term "sufficiently strong" will, of course, be fully explained below.)

Output: A set of q edge-disjoint paths, $\{P_1, \ldots, P_q\}$ such that P_i connects a_i to b_i.

Phase 1. Split G into two spanning expanders $G_R = (V, E_R)$ and $G_B = (V, E_B)$ such that $E = E_R \cup E_B$ and $E_R \cap E_B = \emptyset$. We require G_R to be a 1-expander, and G_B to be a β'-expander, for some $\beta' > 0$. (The details of this procedure are presented in Section 4.1.)

The steady state distribution of the random walk on G_B is easily seen to be given by

$$\pi(v) = \frac{d_B(v)}{2|E_B|} \qquad v \in V,$$

where $d_B(v)$ denotes the degree of the vertex v in G_B. Our construction guarantees that

$$\frac{1}{2n} \leq \pi(v) \leq \frac{3}{2n} \qquad \text{for all } v \in V. \tag{1}$$

Phase 2. Choose independently (with replacement) according to the distribution $\pi(G_B)$, a multiset of $4q$ vertices in V. Let $R = \{r_1, \ldots, r_{4q}\}$ be the multiset of vertices so chosen.

Phase 3. Select a set $Q \subset R$ of $2q$ vertices, such that every pair of vertices in Q are $\kappa_1 \ln \ln n$ apart from each other, as follows:

$Q \leftarrow \emptyset$
for $i = 1, \ldots, 4q$ **while** $|Q| < 2q$ **do**
  if $\text{dist}(Q, r_i) \geq \kappa_1 \ln \ln n$ **then** $Q \leftarrow Q \cup \{r_i\}$ **fi**
od

If at the end of this procedure $|Q| < 2q$ then **stop**. The algorithm has failed.

Phase 4. Let $S = \{a_1, \ldots, a_q, b_1 \ldots, b_q\}$. Using a flow algorithm in G_R, connect in an arbitrary manner the vertices of S to the vertices of Q by $2q$ edge-disjoint paths. (Except for the edges on these paths, no other edges of G_R are used for the final construction.) If such a flow can not be constructed then **stop**. The algorithm has failed. (This can happen only if G_R did not have sufficient edge expansion.)

Phase 5. Let $\tilde{a}_i$ (resp. $\tilde{b}_i$) be the vertex in Q that was connected to a_i (resp. b_i). For each pair $(\tilde{a}_i, \tilde{b}_i)$ construct $m = (\ln n)^2$ paths, $P_{i,1}, \ldots, P_{i,m}$ connecting $\tilde{a}_i$ to $\tilde{b}_i$, as follows:

for $j = 1, 2, \ldots, m$ **do**

Pick a vertex $x_{i,j}$ according to the distribution $\pi(G_B)$.

Choose a trajectory $W'_{i,j}$ (resp. $W''_{i,j}$) of length $\tau = \kappa_2 \ln n$ that goes from $\tilde{a}_i$ to $x_{i,j}$ (resp. $\tilde{b}_i$ to $x_{i,j}$) in G_B, according to the distribution on trajectories, conditioned on $w_{i,j,0} = \tilde{a}_i$ and $w_{i,j,\tau} = x_{i,j}$. (The distribution for $W''_{i,j}$ is analogous.)

Let $W_{i,j}$ be the walk formed by $W'_{i,j}$ followed by $W''_{i,j}$ reversed.

Reduce $W_{i,j}$ to a path $P_{i,j}$ by removing cycles.

od

(The purpose of the remainder of the algorithm is to find among the set of $q \cdot m$ paths constructed in this phase, a *solution set*, that is, a subset of q edge-disjoint paths, one for each pair $(\tilde{a}_i, \tilde{b}_i)$.)

Phase 6. We shall refer to the set of paths $B_i = \{P_{i,1}, P_{i,2}, \ldots, P_{i,m}\}$ as *bundle* i. The purpose of this phase is to prune from each bundle those paths that go "too close" to the endpoints of other bundles or to each other.

Let $w'_{i,j,t}$ and $w''_{i,j,t}$ denote the t'th vertices of $W'_{i,j}$ and $W''_{i,j}$ respectively. Let $M_{i,j} = \{w'_{i,j,t}, w''_{i,j,t} : t \geq (\kappa_1 - \kappa_3) \ln \ln n\}$.

for $i = 1, 2, \ldots, q$ **do**
  for $j = 1, 2, \ldots, m$ **do**
    (a) **if** $\mathrm{dist}(M_{i,j}, \bigcup_{k<j} M_{i,k}) \leq 2\kappa_3 \ln \ln n$ **then** $B_i \leftarrow B_i \setminus \{P_{i,j}\}$ **fi**
    (b) **if** $\mathrm{dist}(W_{i,j}, Q \setminus \{\tilde{a}_i, \tilde{b}_i\}) < \kappa_3 \ln \ln n$ **then** $B_i \leftarrow B_i \setminus \{P_{i,j}\}$ **fi**
  od
od

(Condition (a) ensures that outside the $(\kappa_1 - \kappa_3) \ln \ln n$ neighborhood of the common endpoints, all paths remaining in B_i are at least $2\kappa_3 \ln \ln n$ apart. Condition (b) ensures that all paths in B_i are at least $\kappa_3 \ln \ln n$ from the endpoints of other bundles.)

Let m_i denote the number of paths left in bundle i for $i = 1, 2, \ldots, q$, and rename the paths such that $B_i = \{P_{i,1}, \ldots, P_{i,m_i}\}$.

Check that for all $i \in [q]$, the number of paths in B_i satisfies $m_i \geq (\ln n)^2/2$. If this does not hold then **stop**. The algorithm has failed.

Phase 7. Let $H = (V_H, E_H)$ be the graph defined by

$$V_H = \Big\{(i,j) \mid i = 1, \ldots, q; j = 1, \ldots, m_i\Big\}$$

and

$$E_H = \Big\{\{(i,j),(i',j')\} \mid i \neq i' \text{ and } P_{i,j} \cap P_{i',j'} \neq \emptyset.\Big\}$$

The i'th *row* of H is the set of vertices $\{(i,j) \mid 1 \leq j \leq m_i\}$. A row represents the bundle of paths associated to a certain pair of endpoints, and a solution set corresponds to an independent set of size q that spans all the q rows of H.

Let Δ_H denote the maximum degree of a vertex in H. If there is an i such that $m_i \leq 8\Delta_H$ then **stop**, the algorithm has failed. (As shown in the

analysis of this phase, the Local Lemma implies that the condition $m_i > 8\Delta_H$ is sufficient for the existence of at least one solution set.)

Optionally, for efficiency reasons, we can arbitrarily delete paths from each bundle until for every i, we have $m_i = 8\Delta_H + 1$.

Phase 8. Let $H' = ([q], E_{H'})$ be the graph on q vertices defined by

$$E_{H'} = \Big\{\{i, i'\} \mid \exists j, j' \text{ s.t. } P_{i,j} \cap P_{i',j'} \neq \emptyset.\Big\}$$

(In other words H' contains an edge from i to i' iff any of the paths from $\tilde{a}_i$ to $\tilde{b}_i$ intersects any of the paths from $\tilde{a}_{i'}$ to $\tilde{b}_{i'}$. Clearly H' can be obtained from H by contracting each row of H to a single vertex.)

If any connected component of H' has size greater than $3 \ln n/(2 \ln \ln n)$ then **stop**. The algorithm has failed.

Phase 9. For each connected component J of H', find by exhaustive search, an independent set in H, of size $|J|$, that spans the rows of H corresponding to the vertices of J. (We checked in Phases 6 and 7 that such a set exists, and we checked in Phase 8 that the components of H' are sufficiently small to ensure that the exhaustive search takes only polynomial time.)

The union of independent sets thus found is independent and spans all the rows of H, and hence corresponds to a solution set.

The final path from a_i to b_i is the union of the paths from a_i to $\tilde{a}_i$, and from b_i to $\tilde{b}_i$ found in Phase 4, and the path from $\tilde{a}_i$ to $\tilde{b}_i$ selected here.

End DisjPaths

Observation: Phase 6 and phase 7 are essential only for the proof and can be omitted while running the algorithm. In this case, the exhaustive search in phase 9 would still take only polynomial time, but with small probability (corresponding to failure in phase 6 or 7) it might not find a solution set. Nevertheless, the running time is likely to be shortened if phase 6 and 7 are run, since the search space is reduced.

4 Analysis of the algorithm

4.1 Splitting Expanders

In this subsection we present an algorithm which partitions the edge set of the input graph into two spanning expanders.

Algorithm Split

Input: An r-regular (α, β, γ)-expander graph $G = (V, E)$. For simplicity we assume that $r = 4s$, for an integer s.

Output: Two spanning β'-expanders $G_R = (V, E_R)$ and $G_B = (V, E_B)$ such that $E = E_R \cup E_B$ and $E_R \cap E_B = \emptyset$. (The constant β' is greater than 1 and will be exposed in the proof.)

1. Using an arbitrary Euler tour, orient the edges of G so that each vertex has indegree and outdegree $2s$.

2. For each vertex v, randomly divide the edges from v into a red set and a blue set, each of size s. Set E_R (resp. E_B) to be the set of red (resp. blue) edges, un-oriented.

End Split

Clearly, our construction guarantees that (1) holds, since in each subgraph every vertex has degree at least s and at most $3s$, and each subgraph has exactly ns edges.

We now analyze the probability that **Split** will produce useful results. We start by defining two functions, H and ψ, on [0,1]:

$$\begin{aligned} H(\gamma) &= ((1-\gamma)^{1-\gamma}\gamma^{\gamma})^{-1}, \\ \psi(\epsilon) &= (1-\epsilon)\ln(1-\epsilon) + \epsilon \end{aligned}$$

(Observe that $\psi(\epsilon) \geq \epsilon^2/2$.)

Let $\text{in}_R(S), \text{out}_R(S)$ refer to $\text{in}(S)$ and $\text{out}(S)$ as applied to the graph G_R.

Theorem 3 *Suppose that G is an (α, β, γ)-expander and let $0 < \epsilon < 1$ be such that*

$$\beta > \frac{2}{\psi(\epsilon)} \gamma^{-1} \ln H(\gamma). \tag{2}$$

For every set $S \subset V$, $|S| \leq |V|/2$, we have

$$\min\{\mathrm{out}_R(S), \mathrm{out}_B(S)\} \geq \min\{\alpha, (1-\epsilon)\beta/2\}\,|S|\,, \tag{3}$$

with probability $1 - o(1)$ as $n \to \infty$.

Proof: We obtain a lower bound for out_R. We consider two cases.

Case 1: $|S| \leq \gamma n$**.** By construction every vertex has degree at least s in G_R. Hence

$$\begin{aligned} s|S| &\leq 2\,\mathrm{in}_R(S) + \mathrm{out}_R(S) \\ &\leq 2\,\mathrm{in}(S) + \mathrm{out}_R(S). \end{aligned} \tag{4}$$

On the other hand, by the definition of G,

$$\begin{aligned} 4s|S| &= 2\,\mathrm{in}(S) + \mathrm{out}(S) \\ &\geq 2\,\mathrm{in}(S) + (3s+\alpha)|S|. \end{aligned} \tag{5}$$

Inequalities (4) and (5) imply

$$\mathrm{out}_R(S) \geq \alpha|S|. \tag{6}$$

Case 2: $\gamma n \leq |S| \leq n/2$**.** Partition $\mathrm{out}(S)$ so that 2 edges are in the same subset if in the Euler orientation they have the same start vertex.

Let there be m such sets, $A_1, \ldots, A_m$, with $|A_i| = k_i \leq 2s$, and $\sum_{i=1}^m k_i = k$, where $k \geq \beta|S|$ by the definition of G. Let Z_i be the number of edges of A_i which are colored red. Clearly the Z_i's are independent. For any $t > 0$ and $k/2 > u > 0$ we have

$$\begin{aligned} &\mathbf{Pr}(Z_1 + \cdots + Z_m \leq k/2 - u) \\ &\quad = \mathbf{Pr}\Big(\exp\Big(-t(Z_1 + \cdots + Z_m - k/2 + u)\Big) \geq 1\Big) \\ &\quad \leq \mathbf{E}\Big(\exp\Big(-t(Z_1 + \cdots + Z_m) - k/2 + u\Big)\Big) \\ &\quad = e^{t(k/2-u)} \prod_{i=1}^m \mathbf{E}(e^{-tZ_i}). \end{aligned}$$

But

$$\begin{aligned}\mathbf{E}(e^{-tZ_i}) &= \binom{2s}{s}^{-1}\sum_{j=0}^{k_i}\binom{k_i}{j}\binom{2s-k_i}{s-j}e^{-tj}\\ &\le \left(1+(e^{-t}-1)\frac{k_i}{2s}\right)^s\\ &\le \exp\Big((e^{-t}-1)k_i/2\Big)\end{aligned}$$

For a proof of the first inequality above see either Hoeffding [14] (Section 6) or Chvátal [9]. (Note that although $Z_1, Z_2, \ldots, Z_m$ are independent, a simple application of Theorem 2 of [14] will not suffice. This is because the Z_i have too large a range. The interested reader can check that one obtains a factor s^{-2} in the exponent of the probability bound.)

Hence

$$\begin{aligned}&\mathbf{Pr}\Big(Z_1+\cdots+Z_m \le k/2-u\Big)\\ &\quad\le \exp\Big(t(k/2-u)+(k/2)(e^{-t}-1)\Big)\end{aligned}\tag{7}$$

Putting $t=-\ln(1-2u/k)$ minimizes the RHS of (7) which then becomes $\exp\Big(-(k/2-u)(\ln(1-2u/k))-u\Big)$. Hence if $u=\epsilon k/2$, then

$$\mathbf{Pr}\Big(Z_1+\ldots+Z_m \le (1-\epsilon)k/2\Big) \le e^{-k\psi(\epsilon)/2}$$

and consequently

$$\mathbf{Pr}\Big(\mathrm{out}_R(S) \le (1-\epsilon)\beta|S|/2\Big) \le e^{-\beta|S|\psi(\epsilon)/2}.$$

Thus

$$\begin{aligned}&\mathbf{Pr}\Big(\text{There exists } |S|\ge\gamma n \text{ such that } \mathrm{out}_R(S)\le(1-\epsilon)\beta|S|/2\Big)\\ &\qquad\qquad\le \sum_{i\ge\gamma n}\binom{n}{i}e^{-\beta k\psi(\epsilon)/2}\end{aligned}\tag{8}$$

Now if $i=\theta n$, for $\theta\ge\gamma$ then $\binom{n}{i}=e^{o(n)}H(\theta)^n$, and the summand, u_i say, on the RHS of (8) is then

$$\exp\Big(n(o(1)+\ln H(\theta)-\beta\theta\psi(\epsilon))/2\Big).$$

Now

$$\theta^{-1}\ln H(\theta) = -\ln\theta + 1 - \frac{\theta}{2} - \frac{\theta^2}{6} - \frac{\theta^3}{12} - \dots$$

clearly decreases with θ and so if β satisfies (2) then u_i is exponentially small. The result follows. $\Box$

Corollary 1 *Suppose that G is an (α,β,γ)-expander. Let $0<\epsilon_0<1$ be the unique solution to*

$$\frac{1-\epsilon}{\psi(\epsilon)} = \frac{\gamma}{\ln H(\gamma)} \tag{9}$$

and let

$$\beta_0 = \frac{2}{\psi(\epsilon_0)}\gamma^{-1}\ln H(\gamma).$$

If $\alpha>1$ and $\beta>\beta_0$ then both G_R and G_B are β'-expanders for some $\beta'>1$, with probability $1-o(1)$.

Proof: The existence of ϵ_0 follows from the fact that the LHS of (9) decreases from ∞ to 0 as ϵ increases from 0 to 1. Plugging $\beta>\beta_0$ in (3) we get that $\beta'>1$. $\Box$

It is fairly easy to apply this to the Ramanujuan graphs of Lubotsky, Phillips and Sarnak [17] and to random regular graphs. It follows from Lemma 2.3 of Alon and Chung [4] that

$$|X| = \delta n \text{ implies } \operatorname{out}(X) \geq r(1-\lambda)(1-\delta)|X|, \tag{10}$$

where λ is the second largest eigenvalue of the transition probability matrix associated with the random walk on G. If G is one of the Ramanujan graphs then $\lambda = 2\sqrt{r-1}/r$ and if G is a large random r-regular graph then $\lambda \approx 2/\sqrt{r}$ (see Friedman, Kahn, and Szemerédi [13]). One can then show that in these cases $\min\{\operatorname{out}_R(S),\operatorname{out}_G(S)\} \geq (r/4-o(1))|S|$ for $|S|\leq |V|/2$, as r grows. (For simplicity take $\gamma=\epsilon=r^{-1/3}$.)

The above ideas can be extended to arbitrary graphs. We need to be able to assert that (i) small sets of vertices, $|S|\leq\gamma n$, contain *few* edges; and that (ii) one can orient the edges so that every vertex has *large* outdegree. Given (ii) we can then randomly split the edges into two sets. It is known (Fenner and Frieze [11], Frank [12]) that the edge set of a graph can be oriented

so that the out-degree of each vertex is at least k iff $|\mu(S)| \geq k|S|$ for all $S \subseteq V$ where $\mu(S) = \{e \in E : e \cap S \neq \emptyset\}$, and that this can be checked in polynomial time. We do not however consider this generalization in this paper.

4.2 Analysis of the Main Algorithm

Let P denote the transition probability matrix of the random walk on G_B, and let $P_{v,w}^{(t)}$ 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

$$P_{v,w}^{(t)} = \pi(w) + O(\lambda^t \sqrt{\pi(w)/\pi(v)}). \tag{11}$$

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

$$\text{out}_B(S) \geq \beta'|S| \qquad \text{for all } S \subseteq V,\ |S| \leq |V|/2, \tag{12}$$

for some *constant* $\beta' > 0$, Sinclair and Jerrum [22] show that (12) implies

$$\lambda \leq 1 - \frac{1}{2}\left(\frac{\beta'}{r}\right)^2$$

We will now explicitly state our claims about the performance of our algorithm. As input, G is an n-vertex, bounded degree, r-regular (α, β, γ)-expander graph where $\alpha > 1$ and $\beta > \beta_0$, with β_0 as in Corollary 1.

Suppose that

$$\kappa > \max\{7, \kappa_1 \ln r, 2 + \kappa_3 \ln r\}, \tag{13}$$

$$\kappa_1 > \frac{4 + 2\kappa_3 \ln r}{\ln \lambda^{-1}} + \kappa_3, \tag{14}$$

$$\kappa_2, \kappa_3 > \frac{3}{\ln \lambda^{-1}}. \tag{15}$$

Theorem 4 *Under the above assumptions with n sufficiently large, given any set of $q = n/(\log n)^\kappa$ disjoint pairs of vertices in G such that $\alpha > 1$ and $\beta > \beta_0$, with high probability our algorithm finds in $o(n^3)$ time, edge-disjoint paths connecting these q pairs.*

In Section 3 we pointed out for each phase the conditions under which it might fail. We now proceed to bound the associated failure probabilities.

Phase 1: The failure probability of this phase is $o(1)$ by Corollary 1. Also the time to carry out the construction is $O(n)$. $\Box$

Phase 3: The $\kappa_1 \ln\ln n$ neighborhood of any vertex contains at most $\nu = r^{\kappa_1 \ln\ln n} = (\ln n)^{\kappa_1 \ln r}$ vertices. Using (1), the probability that r_i is rejected is thus never more than $3q\nu/2n$. Thus the probability that this phase fails is at most

$$\mathbf{Pr}(B(4q, 3q\nu/2n) \geq 2q)$$

and this is $o(1)$ if

$$\kappa_1 \ln r < \kappa, \tag{16}$$

since $q \leq n/(\ln n)^{\kappa}$. It is of course straightforward to carry out this selection in $o(n^2)$ time. $\Box$

Phase 4: A straightforward application of the Max-Flow Min-Cut Theorem shows that success is certain provided that G_R is a β'-expander for some $\beta' > 1$. By Corollary 1 this happens with probability $1 - o(1)$. Furthermore it only takes $o(n^3)$ time to find the required flow as arc capacities are 1 for the arcs of the network. $\Box$

Phase 5: The remainder of the proof relies heavily on the fact that the trajectories $W'_{i,j}$ constructed by our algorithm, have the same distribution (up to negligible factors) as m independent random trajectories of length $\tau = \kappa_2 \ln n$ from $\tilde{a}_i$, the difference being that we pick the endpoint of the trajectory using π instead of $P^{(\tau)}_{\tilde{a}_i,\cdot}$. Using (11), since

$$\kappa_2 > \frac{3}{\ln \lambda^{-1}} \tag{17}$$

we see that

$$|P^{(\tau)}_{v,w} - \pi(w)| = O(n^{-3})$$

for all v, w.

In order to allow us to view the trajectories $W'_{i,j}, W''_{i,j}$ as having *exactly* the same distribution as random trajectories we can imagine generating $W'_{i,j}$ as follows:

(a) Choose $x = x_{i,j}$ according to the distribution $P^{(\tau)}_{\tilde{a}_i,\cdot}$.

(b) Choose a random trajectory $W'_{i,j}$ from $\tilde{a}_i$ to x.

(c) If $\theta(x) = P^{(\tau)}_{\tilde{a}_i,x} - \pi(x) > 0$ then with probability $\theta(x)/P^{(\tau)}_{\tilde{a}_i,x}$ **do**

1. discard $W'_{i,j}$;
2. choose $y \in \Omega^- = \{v : \theta(v) < 0\}$ with probability $\theta(y)/\theta(\Omega^-)$;
3. choose a new random trajectory $W'_{i,j}$ from $\tilde{a}_i$ to y.

It is not hard to see that the endpoint of $W'_{i,j}$ other than $\tilde{a}_i$ is now chosen according to the distribution π. Furthermore, conditioned on (1) - (3) above never being executed, we can view $W'_{i,j}$ as a random walk of length τ from $\tilde{a}_i$. But

$$\begin{aligned}&\mathbf{Pr}((1)-(3) \text{ occur during the algorithm})\\ &= O(qm\max\theta(x)) = O((\ln n)^{2-\kappa}/n) = o(1).\end{aligned}$$

This justifies viewing the $W'_{i,j}, W''_{i,j}$ as unbiased random walks.

The next question to answer is as to how, given $x_{i,j}$, do we compute a random trajectory of length τ from $\tilde{a}_i$ to $x_{i,j}$. This is not difficult.

To simplify notation, suppose we want to compute a random trajectory $W = [u_0 = u, u_1, \ldots, u_t = v]$ of length t from a vertex u to a vertex v. If w is a neighbor of v then

$$\mathbf{Pr}(u_{t-1} = w | u_t = v) = \frac{P^{(t-1)}_{u,w} P_{w,v}}{P^{(t)}_{u,v}}. \tag{18}$$

Thus our algorithm to generate W is to choose w according to (18) and then choose a random trajectory of length $t-1$ from u to w. To compute $P^{(t)}$ we need only compute powers of P. Because G has bounded degree we can compute P^k from P^{k-1} in $O(n^2)$ time. Thus the total time to compute all the trajectories is $O(\tau n^2) = o(n^3)$, for $\kappa > 7$. $\Box$

Phase 6: We prove several intermediate propositions. Our aim is to show that relatively few paths get deleted.

Proposition 1 *Assume that*

$$\kappa_1 \geq \frac{4 + 2\kappa_3 \ln r}{\ln \lambda^{-1} + \kappa_3}. \tag{19}$$

Then with probability $1 - o(1)$ *the number of paths deleted due to condition (a) is* $O(\ln n)$ *simultaneously for each* $i \in [q]$.

Proof: Recall that $M_{i,j} = \{w'_{i,j,t}, w''_{i,j,t} : t \geq (\kappa_1 - \kappa_3) \ln \ln n\}$, and a path $P_{i,j}$ is deleted due to condition (a) if $\text{dist}(M_{i,j}, \bigcup_{k<j} M_{i,k}) \leq 2\kappa_3 \ln \ln n$.

For $t \geq (\kappa_1 - \kappa_3) \ln \ln n$ the probability that $w'_{i,j,t} = v$ is (by equation (11)) $O(\lambda^t + 1/n) = O((\ln n)^{-(\kappa_1 - \kappa_3) \ln \lambda^{-1}}$ for any vertex v. Also the $2\kappa_3 \ln \ln n$ neighborhood of $\bigcup_{k<j} M_{i,k}$ is of size $O((\ln n)^{3 + 2\kappa_3 \ln r})$ and so the probability that $W'_{i,j}$ or $W''_{i,j}$ wander into this neighborhood after $(\kappa_1 - \kappa_3) \ln \ln n$ steps, is only

$$O((\ln n)^{3 + 2\kappa_3 \ln r - (\kappa_1 - \kappa_3) \ln \lambda^{-1}}) = O(1/\ln n),$$

given (19). Thus the number of paths deleted from bundle i is dominated by a binomial random variable $B(N, p)$ with $Np = O(\ln n)$.

The inequality (see, e.g. [7] Theorem I.7)

$$\mathbf{Pr}(B(N, p) \geq aNp) \leq \left(\frac{e}{a}\right)^{aNp} \tag{20}$$

is, for sufficiently large a, enough to verify the proposition. (Take $a = c \ln n/(Np)$ for a sufficiently large c.) $\Box$

Proposition 2 *Assume that*

$$\kappa \geq 2 + \kappa_3 \ln r. \tag{21}$$

Let

$$N_i = \{v \in R - \{\tilde{a}_i, \tilde{b}_i\} : \text{dist}(v, B_i) \leq \kappa_3 \ln \ln n\}$$

Then $|N_i| = O(\ln n)$ *simultaneously for each* $i \in [q]$, *with probability* $1 - o(1)$.

Proof: The size of the $\kappa_3 \ln \ln n$ neighborhood of all the bundles B_i together is $O((\ln n)^{3 + \kappa_3 \ln r})$. The number of vertices in R chosen in this neighborhood is a binomial with mean $O(\ln n)$, given (21). The result follows again by using (20). $\Box$

We can now bound the number of paths deleted from each bundle in Phase 6 due to condition (b). Recall that the vertices of $Q \setminus \{\tilde{a}_i\}$ are at least $\kappa_1 \ln\ln n$ away from $\tilde{a}_i$. Hence, if two paths in a bundle are simultaneously closer than $\kappa_3 \ln\ln n$ to the endpoint of another bundle, then one of them is deleted by condition (a). Thus any $v \in N_i \cap Q$ can lead to the deletion of a single path via condition (b), so almost surely only a total $O(\ln n)$ paths are deleted from each bundle. $\Box$

Phase 7: We start by proving

Proposition 3 *The maximum degree in the graph H (the incidence graph of the paths) satisfies $\Delta_H = O((\ln n)^2/\ln\ln n)$, almost surely.*

Proof: We will show below in the analysis of Phase 8 that with probability $1-o(1)$ the graph H' has maximum component size $O(\ln n/\ln\ln n)$ and so it suffices to prove that with probability $1-o(1)$ for every i,j,k, the trajectory $W'_{i,j}$ meets only $O(\ln n)$ trajectories in the bundle B_k.

Now fix i,j,k. The pruning done in Phase 6 allows us to assume now that $\mathrm{dist}(W'_{i,j}, \{\tilde{a}_k, \tilde{b}_k\})$ is at least $\kappa_3 \ln\ln n$. Consider a trajectory $W'_{k,l}$. The probability that $W'_{k,l}$ meets $W'_{i,j}$ is by (11) of order $O((\ln n)^{2-\kappa_3 \ln \lambda^{-1}}) = O(1/(\ln n))$ provided that

$$\kappa_3 \geq \frac{3}{\ln \lambda^{-1}}. \tag{22}$$

Treating the construction of each $W'_{k,l}$ as an independent trial we see that the expected number of trials in which $W'_{i,j} \cap W'_{k,l} \neq \emptyset$ is $O(\ln n)$. We can now use (20). $\Box$

We now show that if we reach the start of Phase 7 and $m_i > 8\Delta_H$ for each i then we can be sure that there is a set of disjoint paths contained in our bundles. We use the following lemma [10, 23]:

Lovász Local Lemma. *Let $A_1, \ldots, A_N$ be events with dependency graph G_A. Let $\deg(i)$ be the degree of A_i in G_A. If*

$$\begin{aligned} \mathbf{Pr}(A_i) &\leq p, \qquad \text{for all } i, \\ \deg(i) &\leq d, \qquad \text{for all } i, \\ 4pd &< 1, \end{aligned}$$

then

$$\mathbf{Pr}(\bigwedge \bar{A}_i) > 0.$$

Consider the experiment in which a random vertex is chosen from each row of H. The events A_i (the "bad" events) are defined by the choice of 2 vertices joined by an edge. The maximum degree in the dependency graph for the Lovász Local Lemma is $2m\Delta_H$ and each bad event has probability at most $4/m^2$. The Local Lemma now proves easily that our independent set exists, since $m = (\ln n)^2$ and $\Delta_H = O((\ln n)^2/\ln\ln n)$.

(N. Alon has shown in [2] the existence of such spanning independent sets in a similar setting using the same technique. Later he proved in [3] that for m sufficiently large relative to the dependency degree there is a proper m-coloring of the graph so that each row uses m colors. Furthermore, J. Beck has shown that this coloring can be found in polynomial time [6]. However, for our purposes it is necessary to show that the component size does not exceed $O(\log n/\log\log n)$ (see below), hence these more sophisticated algorithms seem unnecessary.) $\Box$

Phase 8: Pair the vertices in $R \setminus Q$ arbitrarily. Connect each of the q new pairs by m random walks as is Phase 5. Define a supergraph $H'' \supset H'$ obtained from H' by adding q additional vertices corresponding to bundles of paths connecting vertices of $R \setminus Q$, and by adding extra edges in an obvious way (we ignore the pruning of Phase 6).

Let $\pi(r)$ be the other end of a bundle that has endpoint r. Let $\tilde{W}_r$ be the set of m random walks of length τ starting at vertex $r \in R$. We claim that if r is fixed and r' is chosen uniformly at random from $R \setminus \{r, \pi(r)\}$, then there exists a constant κ_4 such that the probability that $\tilde{W}_r$ and $\tilde{W}_{r'}$ intersect is bounded by

$$\mathbf{Pr}(\tilde{W}_r \cap \tilde{W}_{r'} \neq \emptyset) \leq \frac{\kappa_4(\ln n)^6}{4n}. \tag{23}$$

To see this, consider a random walk from r' of length τ and assume for the moment that r' is chosen uniformly at random in R. Since r' is thus chosen from the steady state of the random walk, the expected number of vertices of $\tilde{W}_r$ visited by this random walk is $O(m\tau^2/n)$. Summing over all walks in $\tilde{W}_{r'}$ we obtain $O(m^2\tau^2/n)$ as the expected number of visits to $\tilde{W}_r$. The probability of at least one visit is bounded by this expectation. Now if r'

is chosen uniformly at random only within $R \setminus \{r, \pi(r)\}$, the probability of intersection increases at most by a factor of $(1 - 2/|R|)^{-1} = 1 + o(1)$. Thus we have (23).

Each bundle B_i is composed of two sets of m random walks. Hence, for i fixed, and i' chosen uniformly at random in $\{1, \ldots, 2q\} \setminus \{i\}$ we have

$$\mathbf{Pr}(\{B_i, B_{i'}\} \text{ is an edge of } H'') \leq \kappa_4 \frac{(\ln n)^6}{n}. \tag{24}$$

Furthermore, if σ is a random permutation of $[2q]$,

$$\begin{aligned} &\mathbf{Pr}(H'' \text{ contains a component of size } \geq k) \\ &\leq \sum_{\substack{S \subseteq [2q] \\ |S| = k}} \sum_{T \in \Omega_{\sigma(S)}} \mathbf{Pr}(\mathcal{E}_T) = \binom{2q}{k} \sum_{T \in \Omega_{\sigma([k])}} \mathbf{Pr}(\mathcal{E}_T), \end{aligned}$$

where Ω_A denotes the set of trees with vertex set A, and $\mathcal{E}_T$ denotes the event that H'' contains a tree isomorphic to T, under the isomorphism $i \leftrightarrow B_i$. The inequality is immediate because any component of size $\geq k$ must contain a tree of size k, and the equality follows from symmetry.

We can restrict our attention to the range where $k/q < 1/2$. We claim that $\mathbf{Pr}(\mathcal{E}_T) \leq (2\kappa_4(\ln n)^6/n)^{k-1}$. Indeed consider the edges of T in a breadth first search order from some arbitrary root; Assume that we have already explored $l - 1$ edges, and thus l vertices numbered without loss of generality $1, \ldots, l$. The probability that the l'th edge exists is given by the probability that for a certain fixed $i \in \{1, \ldots, l\}$ the bundle B_i intersects B'_i where i' is chosen uniformly at random (via σ) in $\{l + 1, \ldots, 2q\}$. This probability can be proven via essentially the same argument used to derive (23) and (24) to be less than $(1 - (2l)/(2q))^{-1}\kappa_4(\ln n)^6/n$, independently of the existence of previous edges.

Now, since $|\Omega_{\sigma([k])}| = k^{k-2}$ we obtain that

$$\begin{aligned} &\mathbf{Pr}(H'' \text{ contains a component of size } \geq k) \\ &= O\left(\left(\frac{2qe}{k}\right)^k \left(\frac{2\kappa_4(\ln n)^6}{n}\right)^{k-1} k^{k-2}\right) \\ &= O\left(\left(\frac{n}{(\ln n)^6}\left(\frac{4e\kappa_4 q(\ln n)^6}{n}\right)^k\right)\right) \end{aligned}$$

$$= O\left(\frac{n}{(\ln n)^6}\left(\frac{4e\kappa_4}{(\ln n)^{\kappa-6}}\right)^k\right) = o(1)$$

for $\kappa > 7$ and $k \geq k_0 = \ln n/(\ln \ln n)$. □

Phase 9: The execution time of Phase 9, given that there are no large components in H'', is bounded by

$$\frac{n}{(\ln n)^\kappa}((\ln n)^2)^{\ln n/\ln \ln n} = o(n^3).$$

□

5 Random $\mathcal{NC}$ Algorithms

In this section we show that our construction can be done in random $\mathcal{NC}$. To convert **DisjPaths** to a random $\mathcal{NC}$ algorithm we need to modify phases 2 and 3 of the algorithm. We replace them by the following two phases:

Phase 2^*. Each vertex $v \in V$ is included in R with probability $8q\pi_v$ independent of the other vertices.

Phase 3^*. A vertex $u \in R$ is in Q if no vertex in its $\kappa_1 \ln \ln n$ neighborhood is in R.

We now consider each phase in turn.

- **Phase 1:** The algorithm **Split** is in $\mathcal{NC}$ since computing an Euler Path is in $\mathcal{NC}$ [5].
- **Phase 2^* and 3^*:** With probability $1 - o(1)$, R has at least $4q$ vertices. The probability that a vertex in R has another vertex in R in its $\kappa_1 \ln \ln n$ neighborhood is smaller than $1/2$, thus with probability $1 - o(1)$, Q has at least $2q$ vertices. The fact that Q might have more than $2q$ vertices does not matter since the flow algorithm gives an integer solution, and only $2q$ vertices in Q will participate in the flow.
- **Phase 4:** Flow with unit capacities is in Random $\mathcal{NC}$ [15, 18].

- **Phase 5,6, and 7:** By attaching one processor to each of the $q(\ln n)^2$ paths used in the algorithm, all these phases can be computed in $O(\ln n)$ time.
- **Phase 8:** Computing connected components is in $\mathcal{NC}$.
- **Phase 9:** Observe that there are no more than $n/(\ln n)^\kappa$ components, and with high probability there are no more than

$$((\ln n)^2)^{\ln n/\ln\ln n} = n^2$$

choices of paths for each component. Given a possible choice, it can be checked by one processor in $O(\ln^2 n)$ steps. Thus, phase 9 can be computed by $o(n^3)$ processors in $O(\ln^2 n)$ parallel steps.

References

[1] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6:83–96, 1986.

[2] N. Alon. The linear arboricity of graphs. *Israel Journal of Mathematics*, 62:311–325, 1988.

[3] N. Alon. The strong chromatic number of a graph. *Random Structures and Algorithms*, 3:1–8, 1992.

[4] N. Alon and F. R. K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 72:15–19, 1989.

[5] B. Awerbuch, A. Israeli, and Y. Shiloach. Finding Euler circuits in logarithmic parallel time. In F. P. Preparata, editor, *Advances in Computing Research 4: Parallel and Distributed Algorithms*, pages 69–78. JAI Press, Greenwich, CT, 1987.

[6] J. Beck. An algorithmic approach to the Lovász Local Lemma I. *Random Structures and Algorithms*, 2:343–365, 1991.

[7] B. Bollobás. *Random Graphs.* Academic Press, 1985.

[8] A. Z. Broder, A. M. Frieze, and E. Upfal. Existence and construction of edge-disjoint paths on expander graphs In *Proceedings of 24th Annual ACM Symposium on Theory of Computing*, pages 140–149, 1992.

[9] V. Chvátal. Probabilistic methods in graph theory. *Annals of Operations Research*, 1:171–182, 1984.

[10] 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*, pages 609–627. North Holland, 1975.

[11] T. I. Fenner and A. M. Frieze. On the connectivity of random m-orientable graphs and digraphs. *Combinatorica*, 2:347–359, 1982.

[12] A. Frank and A. Gyarfas. How to orient the edges of a graph? *Colloq. Math. Soc. J. Bolyai*, 18:353–364, 1978.

[13] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue in random regular graphs. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 587–598, 1989.

[14] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[15] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6:35–48, 1986.

[16] T. Leighton and B. Maggs. Expanders might be practical: Fast algorithms for routing around faults in multibutterflies. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 264–274, 1990.

[17] A. Lubotsky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[18] K. Mulmuley, V. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[19] D. Peleg and E. Upfal. Constructing disjoint paths on expander graphs. *Combinatorica*, 9:289–313, 1989.

[20] D. Peleg and E. Upfal. The token distribution problem. *SIAM Journal of Computing*, 18:229–243, 1989.

[21] N. Robertson and P. D. Seymour. Graph minors-XIII: The disjoint paths problem. To appear.

[22] A. Sinclair and M. Jerrum. Approximate counting, uniform generation, and rapidly mixing Markov chains. *Information and Computation*, 82:93–133, 1989.

[23] J. Spencer. *Ten Lectures on the Probabilistic Method*. SIAM, 1987.

[24] E. Upfal. An $o(\log n)$ deterministic packet routing scheme. In *Proceedings of 21st Annual ACM Symposium on Theory of Computing*, pages 241–250, 1989.