# EXISTENCE AND CONSTRUCTION OF EDGE DISJOINT PATHS ON EXPANDER GRAPHS

by

Andrei Z. Broder
DEC Systems Research Center
130 Lytton Avenue,
Palo Alto, CA USA

and

Alan Frieze
Department of Mathematics
Carnegie Mellon University
Pittsburgh, PA U.S.A.

and

Eli Upfal
IBM Almaden Research Center
San Jose, CA USA

and

Department of Mathematics
Weizmann Institute of Science
Rehovot, Israel

Research Report No. 92–141 2
January, 1992

**Output:** A set of $K$ edge disjoint paths, $\{P_1, \dots, P_K\}$ 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 $\beta'$-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

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

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

**Phase 3.** Select a set $Q \subset R$ of $2K$ 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 $i = 1, \dots, 4K$ while $|Q| < 2K$ do
    if dist$(Q, r_i) \leq \kappa_1 \ln \ln n$ then $Q \leftarrow Q \cup \{r_i\}$ fi
od
```

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

**Phase 4.** Let $S = \{a_1, \dots, a_K, b_1 \dots, b_K\}$. Using a flow algorithm in $G_R$, connect in an arbitrary manner the vertices of $S$ to the vertices of $Q$ by $2K$ 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, $P_{i,1}, \dots, P_{i,m}$ 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 $\tau = \kappa_3 \ln n$
      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 $K \cdot m$ paths constructed in this phase, a *solution set*, that is, a subset of $K$ edge-disjoint paths, one for each pair $(\tilde{a}_i, \tilde{b}_i)$.)

**Phase 6.** (This phase is only needed for the proof and can be dispensed with in the algorithm.)

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 $w'_{i,j,t}, w''_{i,j,t}$ denote the $t$'th vertices of $W'_{i,j}, W_{i,j}$"" respectively. Let $M_{i,j} = \{w'_{i,j,t}, w''_{i,j,t} : t \geq \kappa_1 \ln \ln n\}$.

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

(Condition (a) ensures that outside the $\kappa_1 \ln \ln n$ 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 $i = 1, 2, \ldots, K$, and rename the paths such that $B_i = \{P_{i,1}, \ldots, P_{i,m_i}\}$.

Check that for all $i \in [K]$, 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

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

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 $K$ that spans all the $K$ rows of $H$.

Let $\Delta_H$ 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. The condition $m_i > 8\Delta_H$ is sufficient for the existence of at least one such independent set. This follows from an application of the local lemma and explains the relevance of Phase 6 which is needed in *the proof* that $\Delta_H$ is not too large. More details are given below.

**Phase 8.** Let $H' = ([K], E_{H'})$ be the graph on $K$ 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 Phase 7 that such sets 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**

# 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 $(\alpha, \beta, \gamma)$-expander graph $G = (V, E)$. For simplicity we assume that $r = 4s$, for an integer $s$.

**Output:** Two spanning $\beta'$-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 $\beta'$ 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 $s$.

**End Split**

It should be clear that our construction guarantees that (1) holds.

We now analyze the probability that **Split** will produce useful results. We start by defining two functions, $H$ and $\psi$, 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 $\text{in}_R, \text{out}_R$ refer to in and out as applied to $G_R$.

**Theorem 3** *Suppose that $G$ is an $(\alpha, \beta, \gamma)$-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*

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

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

*Proof:* We obtain a lower bound for $\text{out}_R$. We consider two cases.

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

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

On the other hand

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

Inequalities (4) and (5) imply

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

**Case 2:** $\gamma n \leq |S| \leq n/2$. Partition 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

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

For a proof of the first inequality see either Höeffding [10] (Section 6) or Chvátal [5].

Hence

$$\mathbf{Pr}\Big(Z_1 + \cdots + Z_m \leq k/2 - u\Big) \leq \exp\Big(t(k/2-u) + (k/2)(e^{-t}-1)\Big) \tag{7}$$

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

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

and consequently

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

Thus

$$\mathbf{Pr}\Big(\exists|S| \ge \gamma n : \mathrm{out}_R(S) \le (1-\epsilon)\beta|S|/2\Big) \le \sum_{k\ge\gamma n} \binom{n}{k} e^{-\beta k\psi(\epsilon)/2} \tag{8}$$

Now if $k = \theta n$, for $\theta \ge \gamma$ then

$$\binom{n}{k} = e^{o(n)}H(\theta)^n$$

and the summand, $u_k$ say, on the RHS of (8) is then

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

Now

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

clearly decreases with $\theta$ and so if $\beta$ satisfies (2) then $u_k$ is exponentially small. The result follows. □

**Corollary 1** *Suppose that $G$ is an $(\alpha,\beta,\gamma)$-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 $\beta'$-expanders for some $\beta' > 1$, with probability $1 - o(1)$.*

*Proof:* The existence of $\epsilon_0$ follows from the fact that the LHS of (9) decreases from $\infty$ to 0 as $\epsilon$ increases from 0 to 1. □

It is fairly easy to apply this result to the Ramanujan graphs of Lubotsky, Phillips and Sarnak [13] and to random regular graphs.

It follows from Lemma 2.3 of Alon and Chung [2] that

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

where $\lambda$ 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 [9]). One can then show that in these cases $\min\{\mathrm{out}_R(S), \mathrm{out}_G(S)\} \geq (r/4 - o(1))\,|S|$ 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 [7], Frank [8]) 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 $P^{(t)}_{v,w}$ denote the probability that the walk is at $w$ at step $t$ given that it started at $v$. Let $\lambda$ be the second largest eigenvalue of $P$. (All eigenvalues of $P$ are real.) It is known that

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

To ensure rapid convergence we will need $\lambda \leq 1 - \epsilon$ for some *constant* $\epsilon > 0$. This is achieved if

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

for some *constant* $\beta' > 0$. For instance Sinclair and Jerrum [15] 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 $(\alpha, \beta, \gamma)$-expander graph where

$$\alpha > 1, \beta > \beta_0,$$

($\beta_0$ as in Corollary 1.)

Suppose that

$$\begin{aligned} \kappa &> \max\{7, \kappa_1 \ln r, 2 + \kappa_3 \ln r\}, & (13)\\ \kappa_1 &> \frac{4 + 2\kappa_3 \ln r}{\ln \lambda^{-1}}, & (14)\\ \kappa_2, \kappa_3 &> \frac{3}{\ln \lambda^{-1}}, & (15) \end{aligned}$$

**Theorem 4** *Under the above assumptions with $n$ sufficiently large, given any set of $K = n/(\log n)^{\kappa}$ 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 $K$ 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 $O(n)$. □

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

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

and this is $o(1)$ if

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

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

**Phase 4:** A straightforward application of the Max-Flow Min-Cut Theorem shows that success is certain provided that $G_R$ is a $\beta'$-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. □

**Phase 5:**

The remainder of the proof relies heavily on the fact that the trajectories $W'_{i,j}$ have the same distribution (up to negligible factors) as $m$ independent random trajectories of length $\tau$ from $\tilde{a}_i$. The difference being that we pick the endpoint of the trajectory using $\pi$ instead of $P^{(\tau)}_{\tilde{a}_i,\cdot}$. Using (11) we see that this since

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

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

for all $v, w$.

In order to to allow us to think of 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 $\theta(x)$ **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 $\pi$. Furthermore if (1) - (3) above are never executed then we can view $W'_{i,j}$ as a random walk of length $\tau$ from $\tilde{a}_i$. But

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

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

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

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

If $w$ is a neighbour 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 $O(Km\tau n^2) = o(n^3)$ with our current best value for $\kappa$. □

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

**Proposition 1** *Assume that*

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

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

*Proof:* For $t \geq \kappa_1 \ln\ln n$ the probability that $w'_{i,j,t} = v$ is $O(\lambda^t + 1/n)$ 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 $\kappa_1 \ln\ln n$ steps, is only

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

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

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

is, for sufficiently large $a$, enough to verify the proposition. □

**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* $i \in [K]$*, with probability* $1 - o(1)$.

*Proof:* The size of the $\kappa_3 \ln\ln n$ neighborhood of any $B_i$ 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). □

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 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. □

**Phase 7:**

**Proposition 3** *With probability* $1 - o(1)$ $\Delta_H = O((\ln n)^2 / \ln\ln n)$.

*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 $\text{dist}(W'_{i,j}, \{\tilde{a}_k, \tilde{b}_k\})$ 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). □

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 [6, 16]:

**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*

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

*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 $\Delta_H = O((\ln n)^2/\ln\ln n)$. □

**Phase 8:** Define a supergraph $H'' \supset H'$ obtained from $H'$ by adding $K$ additional vertices corresponding to bundles of paths from the vertices of $R \setminus Q$, and by adding extra edges in an obvious way.

Observe first that if two indices $i, i'$ are chosen prior to construction of the bundles then there exists a constant $\kappa_4$ such that

$$\mathbf{Pr}(\{B_i, B_{i'}\} \text{ is an edge of } H'') \leq \kappa_4 \frac{(\ln n)^6}{n}. \tag{23}$$

To see this, consider a random walk from $\tilde{a}_{i'}$ of length $\tau$. Since $\tilde{a}_{i'}$ is chosen independently of $B_i$ and from the steady state of the random walk, the expected number of vertices of $B_i$ visited is $O(m\tau^2/n)$. Summing over all walks in $B_{i'}$ we obtain $O(m^2\tau^2/n)$ as the expected number of visits to $B_i$. Since the probability of at least one visit is bounded by this expectation we have (23).

Furthermore

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

where $\Omega_S$ denotes the set of trees with vertex set $S$, and $\mathcal{E}_T$ denotes the event that $H''$ contains the tree corresponding to replacing $i$ by $B_i$ in $T$. The inequality is

immediate because any component of size $\geq k$ must contain a tree of size $k$, and the equality follows from symmetry.

It follows from (23) that $\mathbf{Pr}(\mathcal{E}_T) \leq (\kappa_4(\ln n)^6/n)^{k-1}$. (Consider the edges of $T$ in a breadth first search order from some arbitrary root; the existence of each edge is clearly probabilistically independent of the existence of previous edges.) Since $|\Omega_{[k]}| = k^{k-2}$ we obtain that

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

for $\kappa \geq 7$ and $k \geq k_0 = 3\ln n/(2\ln\ln n)$. The expected execution time of Phase 8, given there are no large components in $H''$, is by the above

$$O\left(\sum_{k=1}^{k_0} \frac{n}{(\ln n)^6}\left(\frac{2e\kappa_4}{\ln n}\right)^k m^k\right) = o(n^3).$$

□

This completes the proof of Theorem 4

# 5 Random $\mathcal{NC}$ Algorithms

Algorithm **Split** is clearly in $\mathcal{NC}$ since computing an Euler Path is in $\mathcal{NC}$ [3].

To convert the algorithm **DisjPaths** to a random $\mathcal{NC}$ algorithm we need to modify steps 2 and 3 of the algorithm. We replace them by the following two steps:

1. Each vertex $v \in V$ is included in $R$ with probability $8K\pi_v$ independent of the other vertices.

2. A vertex $u \in R$ is in $Q$ if no vertex in its $\kappa_1 \ln\ln n$ neighborhood is in $R$.

With probability $1 - 1/n^2$, $R$ has at least $4K$ 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 $1 - 1/n^2$, $Q$ has at least $2K$ vertices. The fact that $Q$ might have more than $2K$ vertices does not matter since the flow algorithm gives an integer solution, and only $2K$ vertices in $Q$ will participate in the flow.

Flow with unit capacities is in Random $\mathcal{NC}$, [11], [14], thus step 4 of the algorithm is in Random $\mathcal{NC}$.

By attaching one processor to each of the $K(\ln n)^2$ paths, steps 5–7 can be computed in $O(\ln n)$ time. Step 8 is in $\mathcal{NC}$ since computing connected components is in $\mathcal{NC}$.

To compute step 9 we observe that there are no more than $n$ components, and with probability $1 - 1/n^2$, there are no more than $(12(\ln n)^2)^{\ln n/\ln\ln n} = O(n^2)$ choices of paths for each component. Given a possible choice, it can be checked by one processor in $O(\log^2 n)$ steps. Thus, step 9 can be computed by $O(n^3)$ processors in $O(\log^2 n)$ parallel steps.

# References

[1] N. Alon. A Parallel Algorithmic Version of the Local Lemma. *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, San Juan, Puerto Rico, 1991.

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

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

[4] J. Beck. An Algorithmic Approach to the Lovász Local Lemma. To appear in *Random Structure and Algorithms*, 1991.

[5] V. Chvátal Probabilistic methods in graph theory In *Stochastics and Optimization* (F.Archetti and F.Maffioli Eds.), *Annals of Operations Research*, 1:171–182, 1984.

[6] P. Erdös and L. Lovász. Problems and Results on 3-Chromatic Hypergraphs and Some Related Questions. In *Infinite and Finite Sets*, (A. Hajnal et. al. eds), Colloq. Math. Soc. J. Bolyai 11, North Holland, 1975, pp. 609-627.

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

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

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

[10] W. Hoeffding. Probability Inequalities for Sums of Bounded Random variables. In *Journal of the American Statistical Association*, 58:13–30, 1963.

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

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

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

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

[15] A.Sinclair and M.Jerrum Approximate Counting, Uniform Generation, and Rapidly Mixing Markov Chains. *Information and Computation*, 82:93–133, 1989.

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

[17] D. Peleg and E. Upfal. Construction Disjoint Paths on Expander Graphs. *Combinatorica*, 9:289–313, 1989.

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

[19] N.Robertson and P.D.Seymour Graph minors-XIII: The disjoint paths problem. *to appear*

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