Arc-Disjoint Paths in Expander Digraphs

Tom Bohman* and Alan Frieze†

Department of Mathematical Sciences,
Carnegie Mellon University
Pittsburgh PA 15213.

Abstract

Given a digraph $D = (V, A)$ and a set of κ pairs of vertices in V, we are interested in finding, for each pair (x_i, y_i), a directed path connecting x_i to y_i such that the set of κ paths so found is arc-disjoint. For arbitrary graphs the problem is $\mathcal{NP}$-complete, even for $\kappa = 2$.

We present a polynomial time randomized algorithm for finding arc-disjoint paths in an r-regular expander digraph D. We show that if D has sufficiently strong expansion properties and the degree r is sufficiently large then *all* sets of $\kappa = \Omega(n/\log n)$ pairs of vertices can be joined. This is within a constant factor of best possible.

1 Introduction

Given a (graph) digraph $D = (V, A)$ with n vertices and a set of κ pairs of vertices in V, we are interested in finding, for each pair (x_i, y_i), a (undirected) directed path connecting x_i to y_i, such that the set of κ paths so found is (edge) arc-disjoint. This is a classical problem in graph theory. See Frank [7] for a survey and Chapter 9.2 of the recent book on digraphs by Bang-Jensen and Gutin [2].

For undirected graphs, the related decision problem is in $\mathcal{P}$ for fixed κ – Robertson and Seymour [22], but is $\mathcal{NP}$-complete if κ is part of the input. For digraphs the situation is seemingly much worse. Fortune, Hopcroft and Wylie [6] showed that the related decision problem is $\mathcal{NP}$-complete, even when $\kappa = 2$.

For undirected graphs there have been positive results in the case of expanders. Peleg and Upfal [21] presented a polynomial time algorithm for the case where D is a (sufficiently strong) bounded degree expander graph and $\kappa \leq n^{\epsilon}$ for a small constant ϵ that depends on the expansion property of the graph. This result has been improved and extended by

*Supported in part by a grant from the NSA.

†Supported in part by NSF grant CCR-9818411.

Broder, Frieze, and Upfal [4, 5], Frieze [8, 9], Leighton and Rao [17] and Leighton, Rao and Srinivasan [18, 19]. In particular Frieze [9] showed that if D has sufficiently strong *edge* expansion properties and r is sufficiently large then *all* sets of $\kappa = \Omega(n/\log n)$ pairs of vertices can be joined. This is within a constant factor of a simple upper bound. The purpose of this paper is to extend this result to digraphs.

In this paper we discuss r-regular digraphs. A digraph D is r-regular if every vertex has in-degree and out-degree r. Let d_μ be the median distance between pairs of vertices in the digraph D which has m arcs. Clearly, there exists a collection of $O(m/d_\mu)$ pairs of vertices that cannot be connected by arc-disjoint paths because such a collection of paths would require more arcs than all the arcs available. In the case of an r-regular expander, this absolute upper bound on κ is $O(n/\log n)$ (assuming r is independent of n). We show that if D has sufficiently strong *arc* expansion properties and r is sufficiently large then *all* sets of $\kappa = \Omega(n/\log n)$ pairs of vertices can be joined. This therefore, is within a constant factor of the optimum. The precise definition of "sufficiently strong" is given after the theorem.

Theorem 1 *Let $D = (V, A)$ be an n-vertex, r-regular digraph. Suppose that D is a sufficiently strong arc expander. Then there exist $\epsilon_1, \epsilon_2 > 0$ such that D has the following property: For all sets of pairs of vertices $\{(x_i, y_i) \mid i = 1, ..., \kappa\}$ satisfying*

(i) $\kappa = \lceil \epsilon_1 rn/\log n \rceil$, *and*
(ii) For each vertex v, $|\{i : x_i = v\}|, |\{i : y_i = v\}| \leq \epsilon_2 r$

there exist arc-disjoint paths in D, each of length $O(\log n)$, joining x_i to y_i, for each $i = 1, 2, \ldots, \kappa$. Furthermore, there is a polynomial time randomized algorithm for constructing these paths. The constants ϵ_1, ϵ_2 depend only on certain expansion parameters α, β, γ defined below. They do not depend on n or r. (For example conditions (2) with $\epsilon = \alpha$, (3), (9) and (12) suffice).

Remark 1 *The algorithm is similar to the algorithm of [9]. The difficulty in moving from graphs to digraphs has been with that part of the algorithm for graphs which was based on the rapid mixing of a random walk on expanders. Random walks on digraphs are not necessarily time reversible and the steady state can be hard to determine. We therefore abandoned this approach and replaced it with a different random choice of path (in part, this random choice for digraphs uses the multicommodity flow results of Leighton and Rao [16]).*

It will be observed that this new algorithm can substitute for that given in [9].

Remark 2 *If D has sufficiently strong* vertex *expansion properties then we can take* $\kappa = \lceil \epsilon_1 rn/\log_r n \rceil$ *– see Remark 4 below.*

Remark 3 *It is perhaps worth remarking that regularity is not crucial to the result. One can easily extend the results to digraphs where in- and out-degrees are constrained to be in the interval $[r, ar]$ where $a \geq 1$ is an absolute constant. All we need is for r to be sufficiently large. The really crucial property is strong expansion.*

1.1 Preliminaries

In this section we state the definitions for the expanders we work with here, make some preliminary observations about such expanders, and make precise the notion of 'sufficiently strong' expansion needed for Theorem 1. We begin with some notational conventions.

Let $D = (V, A)$ be a digraph and let $n = |V|$. For $S, T \subseteq V$ let $A^+_D(S,T)$ be the set of arcs with tail in S and head in T; that is,

$$A^+(S,T) = A^+_D(S,T) = \{(u,v) \in A \mid u \in S, v \in T\} \quad \text{and} \quad d^+(S,T) = |A^+(S,T)|.$$

We define $A^-(S,T)$ similarly (i.e $A^-(S,T) = A^+(T,S)$). We set

$$A^+(S) = \{(u,v) \in A \mid u \in S, v \notin S\} \quad \text{and} \quad d^+(S) = |A^+(S)|.$$

So, for example, we have $A^+(S) = A^+(S, V \setminus S)$. We define $A^-(S)$ and $d^-(S)$ analogously. Throughout the paper, when $\star$ is used as a subscript or superscript, it stands for + or -. We abbreviate $A^\star(\{v\},T)$ to $A^\star(v,T)$. Thus, for $v \in V$, $d^+_D(v)$ and $d^-_D(v)$ denote the out-degree and in-degree of v in D.

We now have the notation necessary to introduce expansion. We define expanders in terms of arc expansion (a weaker property than vertex expansion). For $S \subseteq V$ let $\Phi^\star_S = d^\star(S)/|S|$. The (arc-)expansion $\Phi = \Phi(D)$ of D is defined by

$$\Phi = \min_{\substack{S \subseteq V \\ |S| \leq n/2}} \min\{\Phi^+_S, \Phi^-_S\}.$$

A digraph $D = (V, A)$ is a θ-expander, if for every set $S \subseteq V$, $|S| \leq n/2$, we have $d^\star(S) \geq \theta|S|$; in other words, D is a θ-expander if $\Phi(D) \geq \theta$. An r-regular digraph $D = (V, A)$ is called an (α, β, γ)-expander if for every set $S \subseteq V$

$$d^\star(S) \geq \begin{cases} (1-\alpha)r|S| & \text{if } |S| \leq \gamma n \\ \beta r|S| & \text{if } \gamma n < |S| \leq n/2 \end{cases}$$

We naturally assume that $\beta < 1 - \alpha$. By "sufficiently strong" in Theorem 1, we mean that β, γ are arbitrary and α is sufficiently small; in particular, we assume that conditions (2) with $\epsilon = \alpha$, (3), (9) and (12) hold. We also assume throughout that r and n are sufficiently large (but r is *not* a function of n). We have made no real attempt to optimize constants. Such digraphs exist; in particular, random regular digraphs are usually (α, β, γ)-expanders. (See discussion in [4] for the corresponding notion in undirected graphs.)

We conclude this section with some preliminary observations about expanders. Since $|A^\star(S,S)| + d^\star(S) = r|S|$ we see that, putting $in(S) = |A^+(S,S)| = |A^-(S,S)|$, in an (α, β, γ)-expander

$$in(S) \leq \alpha r|S| \qquad \text{when } |S| \leq \gamma n. \tag{1}$$

For a digraph $\Delta = (V', A')$ and a set $S \subseteq V'$ we define its out-neighbor set $N^+_\Delta(S)$, as

$$N^+_\Delta(S) = \{w \notin S : \exists v \in S \text{ such that } (v,w) \in A'\}.$$

Similarly, the in-neighbor set of S, $N^-_\Delta(S)$ is given by

$$N^-_\Delta(S) = \{w \notin S : \exists v \in S \text{ such that } (w,v) \in A'\}.$$

Lemma 1 *Suppose that $D = (V, A)$ is an (α, β, γ)-expander and that $D' = (V', A')$ is a sub-digraph of D of expansion at least θr where $\theta > \alpha$. Suppose $S \subseteq V'$. If $|S| \le \frac{\gamma\alpha}{\theta} n$ then*

$$|N^\star_{D'}(S)| \ge \frac{\theta - \alpha}{\alpha}|S|.$$

Proof: Suppose that $|S| \le \frac{\gamma\alpha}{\theta} n$ and $T = N^\star_{D'}(S)$ satisfies $|T| < \frac{\theta-\alpha}{\alpha}|S|$. Then

$$|S \cup T| < \left(1 + \frac{\theta - \alpha}{\alpha}\right)|S| = \frac{\theta}{\alpha}|S| \le \gamma n.$$

But, $S \cup T$ contains at least

$$\theta r|S| > \theta r \left(1 + \frac{\theta - \alpha}{\alpha}\right)^{-1}|S \cup T| = \alpha r|S \cup T|$$

arcs, which contradicts (1). $\square$

2 The algorithm

The input to our algorithm is a sufficiently strong (α, β, γ)-expander digraph D and a set of pairs of vertices $\{(x_i, y_i) \mid i = 1, ..., \kappa\}$ satisfying the premises of Theorem 1. The output is a set of κ arc-disjoint paths, $P_1, \ldots, P_\kappa$ such that P_i connects x_i to y_i.

The algorithm has three phases. We begin in Phase 0 by splitting our graph into 13 arc-disjoint expanders $D_i = (V, A_i)$, $1 \le i \le 13$. These graphs will be used for various purposes in Phases 1 and 2, and each path we construct will be a union of paths in the D_i's. Phase 1 consists mainly of applications of GenPaths, an algorithm that uses the expander property to naively connect pairs of vertices with paths of length $O(\log n)$ one at a time, deleting the arcs from a path as soon as it is used. Of course, this deletion of arcs may quickly destroy the expander property. To compensate for this problem, GenPaths 'shrinks' the expanders in which it finds paths. The real work of GenPaths is in keeping as many vertices as possible 'connected' to these shrinking expanders. Those pairs of vertices that are not connected with paths in Phase 1 (i.e. those pairs that contain a vertex whose connection with one of the 'shrinking expanders' is lost) are handled in Phase 2. There are $O\left(\frac{n}{\log^4 n}\right)$ such pairs. Loosely speaking, Phase 2 uses the multicommodity flow algorithm of Leighton and Rao to give a distribution on paths connecting the remaining pairs such that **whp** paths chosen at random with respect to this distribution are arc disjoint.

2.1 Phase 0.

We need an algorithm for splitting a (α, β, γ)-expander digraph into 13 expander digraphs. Algorithms for splitting undirected graphs are given in [4] or [10]. They are easily adapted to digraphs and we outline an adaptation of the algorithm of [10] in Appendix A. In the appendix we prove:

Theorem 2 *Suppose we have*

$$\frac{r}{\log r} \geq 91\epsilon^{-2} \text{ and } \beta \geq 65\epsilon^{-2} r^{-1} \log 2er \tag{2}$$

and that G is an r-regular (α, β, γ)-expander, r constant. Then there is a randomized polynomial time algorithm which with probability at least $1 - \delta$ constructs $A_1, A_2, \ldots, A_{13}$ such that the arc-expansion Φ_i of $D_i = (V, A_i)$ satisfies

$$\Phi_i \geq (1 - \epsilon)\frac{\Phi}{13} - (\alpha + 2\epsilon)\, r,$$

for $i = 1, 2, \ldots, 13$.

This theorem is only useful if Φ is at least a constant multiple of r and α is sufficiently small. This is the case discussed in this paper. The algorithm runs in $O(n^2 \ln n \log \delta^{-1})$ expected time. There is not enough time to verify that the algorithm succeeds. Instead, we simply assume it has and repeat the split if we fail to find the required paths.

We apply the algorithm of Theorem 2 with $\alpha = \epsilon$ and assume that

$$\beta > 156\alpha. \tag{3}$$

Setting

$$\beta_0 = \frac{\beta}{13} - 4\alpha > 8\alpha > 0, \tag{4}$$

each D_i satisfies

$$\Phi_i = \Phi(D_i) \geq \beta_0 r, , \qquad \text{and} \tag{5}$$

$$\beta_0 r \leq d_i^\star(v) < r, \qquad \text{for all } v \in V. \tag{6}$$

2.2 Phase 1.

Phase 1 uses expanders D_1 through D_6. The centerpiece of Phase 1 is the algorithm GenPaths that connects a large collection of *random* pairs of vertices in an expander with arc-disjoint paths. Since the pairs x_i, y_i are arbitrary, GenPaths cannot be applied to them directly; we must reduce the problem of connecting the x_i's to the y_i's to the problem of connecting random pairs.

In order to produce random pairs we introduce three random sets of κ vertices: $\tilde{X}$, $\tilde{Y}$ and Z. In the Initialization step of Phase 1, a network-flow technique is used to find a collection of arc-disjoint paths $\mathcal{P}^1 = \{P_i^1 : i = 1, \ldots, \kappa\}$ from $X = \{x_1, \ldots, x_\kappa\}$ to $\tilde{X}$ in expander D_1 such that the path P_i^1 starts at x_i for $i = 1, \ldots, \kappa$. It is important to note that we have no control over which element of $\tilde{X}$ is at the end of the path P_i^1 (but there will be one path ending at x for each $x \in \tilde{X}$). This network-flow technique is also used in the Initialization step of Phase 1 to find a collection of arc-disjoint paths $\mathcal{P}^6 = \{P_i^6 : i = 1, \ldots, \kappa\}$ from $\tilde{Y}$ to $Y = \{y_1, \ldots, y_\kappa\}$ in D_6 such that the endpoint of the path P_i^6 is y_i. After the Initialization step, we take a random ordering of $\tilde{X}$: $\tilde{X} = \{\tilde{x}_1, \ldots \tilde{x}_\kappa\}$. Furthermore, we order $\tilde{Y}$ so as to respect the pairing of $\tilde{X}$ and $\tilde{Y}$ that is inherited from the collections $\mathcal{P}^1$ and $\mathcal{P}^6$; that is, we set $\tilde{Y} = \{\tilde{y}_1, \ldots, \tilde{y}_\kappa\}$ so that if the endpoint of P_i^1 is $\tilde{x}_j$ then $\tilde{y}_j$ is the starting point of P_i^6. Thus, it remains to find a collection of arc-disjoint paths connecting the pairs $\{(\tilde{x}_i, \tilde{y}_i) : i = 1, \ldots, \kappa\}$. Unfortunately this sequence of pairs of vertices is *not* truly random. This is a consequence of the fact that the pairing between $\tilde{X}$ and $\tilde{Y}$ is determined by a deterministic process (i.e. knowledge of some of the *pairs* from this collection may bias the distribution on the unknown pairs). We overcome this problem by introducing the third random set $Z = \{z_1, \ldots, z_\kappa\}$. The sequences $\{(\tilde{x}_i, z_i) : i = 1, \ldots, \kappa\}$ and $\{(z_i, \tilde{y}_i) : i = 1, \ldots, \kappa\}$ are perfectly random sequences of pairs of vertices (so long as we view them separately). It remains to connect these two sequences of pairs of vertices with arc-disjoint paths. This is the work of the algorithm GenPaths.

The input to GenPaths is a pair of expanders, D_a and D_b, and a collection of pairs of vertices $\{(v_i, u_i) : i = 1, \ldots, \kappa\}$ that is generated uniformly at random. The output of GenPaths is a collection of arc-disjoint paths from v_i to u_i for $i = 1, \ldots, \kappa$ that use only the arcs from D_a and D_b. We apply GenPaths twice. For the first application we set $D_a = D_2$, $D_b = D_3$ and $\{(v_i, u_i) : i = 1, \ldots, \kappa\} = \{(\tilde{x}_i, z_i) : i = 1, \ldots, \kappa\}$. In the second application we set $D_a = D_4$, $D_b = D_5$ and $\{(v_i, u_i) : i = 1, \ldots, \kappa\} = \{(z_i, \tilde{y}_i) : i = 1, \ldots, \kappa\}$. Now, the expander D_a is used to connect the v_i's to the u_i's with *short* paths one at a time. In order to be sure that such paths exist we must be working with an expander. Therefore we delete some vertices in the course of the algorithm; in other words, this expander shrinks as the algorithm progresses. D_b is used to keep as many vertices as possible connected to the 'shrinking expander' contained in D_a. We should note that these connections also require that D_b be an expander. So, D_b also 'shrinks' in the course of the algorithm. The subroutines Remove and ConnectBack are used by GenPaths.

2.2.1 Initialization

Let $\tilde{X}, \tilde{Y}$ be two randomly chosen κ-subsets of V. We begin by replacing the problem of finding paths from x_i to y_i by that of finding paths from a_i to b_i, where $a_i \in \tilde{X}$ and $b_i \in \tilde{Y}$. Let X denote the set $\{x_1, x_2, \ldots, x_\kappa\}$ and $Y = \{y_1, y_2, \ldots, y_\kappa\}$. We connect X to $\tilde{X}$ via arc-disjoint paths in the digraph D_1 using a network flow. We construct our network as follows

- Each directed arc of D_1 gets capacity 1.
- Each $v \in V$ becomes a source of capacity $|\{i : \ x_i = v\}|$ and each member of $\tilde{X}$ becomes a sink of capacity 1.

Then we find a flow from X to $\tilde{X}$ that satisfies all demands. We can find such a maximum flow with integer values and this decomposes naturally into $|X|$ arc-disjoint paths (together perhaps with some cycles). We connect $\tilde{Y}$ to Y by arc-disjoint paths in a similar manner using D_6

We now have a collection of arc-disjoint paths $\mathcal{P}^1 = \{P_i^1 : i = 1, \ldots \kappa\}$ such that the starting point of P_i^1 is x_i for $i = 1, \ldots, \kappa$ and each member of $\tilde{X}$ is the endpoint of exactly one path from $\mathcal{P}^1$. Furthermore, we have a collection of arc-disjoint paths $\mathcal{P}^6 = \{P_i^6 : i = 1, \ldots \kappa\}$ such that each member of $\tilde{Y}$ is the starting point of exactly one path from $\mathcal{P}^6$ and the end-point of path P_i^6 is y_i for $i = 1, \ldots, \kappa$.

2.2.2 Algorithm GenPaths.

The aim of GenPaths is to join v_i and u_i for $i = 1, 2, \ldots, \kappa$ by a *short* (i.e. of length $O(\log n)$) path in D_a. After constructing a path, we remove its arcs. It is important to ensure that short paths exist. Of course, this would not be a problem if we could ensure that D_a remains an expander throughout. We have to be satisfied with identifying a *dynamically changing large* subgraph $\Delta_a = (V_a, F_a)$ of D_a which is an expander. Initially $\Delta_a = D_a$, and V_a loses vertices as GenPaths progresses. We ensure that Δ_a remains an expander by keeping the degrees of vertices in the Δ_a close to their degree in D_a. This may involve deleting some (low degree) vertices after the construction of a path. We use the routine remove to do this.

If the proposed start vertex v of a walk on Δ_a does not lie in V_a then we try to *connect it back* to V_a by a path in D_b. The terminal endpoint of this walk is denoted by v'. We use a subroutine ConnectBack for this purpose. Similarly, the proposed end vertex u might not lie in V_a. In this case we use ConnectBack to find a path from some $u' \in V_a$ to u in D_b. We do not expect to succeed all the time and our failures are kept in a set L for treatment in Phase 2. The arcs in the paths generated by ConnectBack are deleted from D_b. Since ConnectBack requires that D_b is an expander we will also be working with a second 'shrinking expander' $\Delta_b = (V_b, F_b)$ contained in D_b. This shrinking expander will also be maintained by use of the subroutine Remove.

In the end, the path from v_i to u_i will be a concatenation of up to three separate paths. There will always be a path Q_i from D_a, and there may also be a short walk (or walks) from D_b provided by ConnectBack (these are denoted $W_i^{CB\rightarrow}$ and $W_i^{CB\leftarrow}$, respectively).

Algorithm GenPaths
begin
  $\Delta_t \leftarrow D_t$, $t = a, b$.
  for $i = 1$ **to** κ **do**
  Execute REMOVE(Δ_a)
  Execute ConnectBack$(V_a, v_i, \rightarrow, v_i', i, W_i^{CB\rightarrow})$
  Execute ConnectBack$(V_a, u_i, \leftarrow, u_i', i, W_i^{CB\leftarrow})$
  if $i \notin L$ **then**
    Construct a shortest path Q_i from v_i' to u_i' in Δ_a.
    $P_i \leftarrow (W_i^{CB\rightarrow}, Q_i, W_i^{CB\leftarrow})$
    $\Delta_a \leftarrow \Delta_a \setminus E(P_i)$
  fi
  od
end GenPaths

2.2.3 Subroutine Remove

The purpose of REMOVE is to delete vertices which might prevent a digraph from being an expander. In the course of GenPaths we apply REMOVE to Δ_a and Δ_b. In words, this simple algorithm iteratively removes those vertices whose in/out degree is less than the original in/out degree minus $\beta_0 r/2$. To be precise, at the end of the algorithm we have a graph $\Delta_t = (V_t, F_t)$ where $t \in \{a, b\}$ such that

$$v \in V_t \text{ implies } d^\star_{\Delta_t}(v) \geq d^\star_{D_t}(v) - \beta_0 r/2 \geq \beta_0 r/2. \tag{7}$$

The final inequality in (7) follows from (6). It follows immediately from (7) that for $S \subseteq V_t$ we have

$$d^\star_{\Delta_t}(S) \geq d^\star_{D_t}(S) - \beta_0 r|S|/2 \geq (\Phi_t - \beta_0 r/2)|S|.$$

This implies that, provided neither V_a nor V_b become empty (an issue which we take up in the next section), both Δ_a and Δ_b are expanders throughout Phase 1:

$$\Phi_{\Delta_t} \geq \Phi_t - \beta_0 r/2 \geq \beta_0 r/2 \qquad \text{for } t = a, b. \tag{8}$$

Algorithm Remove(Δ_t)
begin
  $B \leftarrow \{v \in V_t : \; d^+_{\Delta_t}(v) < d^+_{D_t}(v) - \beta_0 r/2 \;$ or $\; d^-_{\Delta_t}(v) < d^-_{D_t}(v) - \beta_0 r/2\}$.
  if $B \neq \emptyset$ **then**
  $A \leftarrow V_t \setminus B$
  $d \leftarrow \max\{\, \max\{d^+_{D_t}(v) - d^+_{\Delta_t}(v, A) \,,\, d^-_{D_t}(v) - d^-_{\Delta_t}(v, A)\} : \; v \in A\}$.
  while $d > \beta_0 r/2$ **do**
    $C \leftarrow \{w \in A : \max\{d^+_{D_t}(w) - d^+_{\Delta_t}(w, A) \,,\, d^-_{D_t}(w) - d^-_{\Delta_t}(w, A)\} \; \geq \beta_0 r/2\}$
    $B \leftarrow B \cup C$
    $A \leftarrow A \setminus C$
    $d \leftarrow \max\{\, \max\{d^+_{D_t}(v) - d^+_{\Delta_t}(v, A) \,,\, d^-_{D_t}(v) - d^-_{\Delta_t}(v, A)\} : \; v \in A\}$.
  od
  $V_t \leftarrow A$
  fi
end remove

2.2.4 Subroutine ConnectBack.

The purpose of ConnectBack is to connect a vertex z to V_a by means of a short walk in D_b. The direction of this walk is determined by the input dir. If dir $=\rightarrow$ then a path from z to V_a is required, and if dir $=\leftarrow$ then a path from V_a to z is needed. If $z \in V_a$ already then ConnectBack does nothing but relabel z as z'. Since $|V \setminus V_a|$ can be of order n (this is discussed below), we must maintain the expander property of Δ_b in order to find short connecting paths. Thus we apply Remove to Δ_b in the course of ConnectBack. Now, those pairs that contain a vertex that lies in $V \setminus (V_a \cup V_b)$ are passed to Phase 2 in the set L. Thus, the long term success of ConnectBack hinges on keeping $V \setminus (V_a \cup V_b)$ small. In fact, this can be viewed as the key point in all of GenPaths.

We keep V_b large by ensuring that the paths we use in Δ_b are spread out; in other words, we avoid using too many paths through any one vertex. This is achieved **whp**; recall that the pairs of vertices $\{(v_i, u_i) : i = 1, \ldots, \kappa\}$ that are the input to GenPaths are assumed to be generated uniformly at random. When a path is needed (i.e when $z \notin V_a$) ConnectBack constructs a collection $\mathcal{W}^{\rm dir}$ of walks in Δ_b. This collection has the following properties.

1. If dir $=\rightarrow$ then every walk in $\mathcal{W}^{\rm dir} = \mathcal{W}^{\rightarrow}$ is a walk from a distinct vertex in $V_b \setminus V_a$ to V_a.

2. If dir $=\leftarrow$ then every walk in $\mathcal{W}^{\rm dir} = \mathcal{W}^{\leftarrow}$ is a walk from V_a to a distinct vertex in $V_b \setminus V_a$.

3. No path in $\mathcal{W}^{\rm dir}$ is longer than $22 \log\log n$.

4. No vertex of D_b lies on more than $240(\log\log n)^2$ paths.

The set of start vertices of the walks in $\mathcal{W}^{\rightarrow}$ is denoted $S_{CB}^{\rightarrow}$ and the set of terminal vertices of the walks in $\mathcal{W}^{\leftarrow}$ is denoted $S_{CB}^{\leftarrow}$. Clearly, $S_{CB}^{\mathrm{dir}} \subseteq V_b \setminus V_a$. The collection of walks will have the additional property

5. $|V \setminus (V_a \cup S_{CB}^{\mathrm{dir}})| \leq \frac{n}{(\ln n)^4}$.

In words, condition 3 says that the paths in $\mathcal{W}^{\mathrm{dir}}$ are short, condition 4 says that the paths are 'spread out' and condition 5 says that very few vertices are left out of the collection.

We emphasis that $\mathcal{W}^{\mathrm{dir}}$ is constructed without use of any information about z. Therefore, z (which was a random vertex to begin with) can be viewed as a vertex chosen uniformly at random *after* the collection $\mathcal{W}^{\mathrm{dir}}$ is constructed. Heuristically, we can think of $\mathcal{W}^{\rightarrow}$ and $\mathcal{W}^{\leftarrow}$ as collections of connecting paths that are updated whenever Δ_a or Δ_b 'shrink,' but we only 'look' at these collections when we need them.

If $z \in S_{CB}^{\mathrm{dir}}$ then we connect z back to V_a by way of the unique path in $\mathcal{W}^{\mathrm{dir}}$that begins at z (if dir $=\rightarrow$) or ends at z (if dir $=\leftarrow$). If z does not lie in S_{CB}^{dir} we put i into L (note that we have either $z = \tilde{x}_i$ or $z = \tilde{y}_i$). Arc disjoint paths for the pairs $(\tilde{x}_i, \tilde{y}_i), i \in L$ are found in Phase 2.

A network flow technique for the construction of the collection of paths $\mathcal{W}^{\mathrm{dir}}$ follows from the proof of Lemma 3, which is given in Section 2.3.3 below.

```
subroutine ConnectBack(V_a, z, dir, z', i, W_CB)
begin
  if z ∈ V_a
    then z' ← z
    else
      Execute remove(Δ_b)
      Contruct W^dir (see Lemma 3 for algorithm).
      if z ∉ S_CB^dir
        then L ← L ∪ {i}
        else
          W_CB ← the unique path in W^dir with start/terminal vertex z
          z' ← terminal/start vertex of W_CB
          Δ_b ← Δ_b \ W_CB
      fi
  fi
end ConnectBack
```

2.3 Analysis of Phase 1

There are three facts concerning Phase 1 that remain to be shown: that the flow needed in the Initialization exists, that V_3 stays large and that at the end of Phase 1 **whp** we have $L = O\left(\frac{n}{(\log n)^4}\right)$.

2.3.1 Initialization

In this subsection we show that if (5) holds and r is sufficiently large then we can find arc-disjoint paths from $\{x_1, \ldots, x_\kappa\}$ to $\tilde{X}$ in D_1 and arc-disjoint paths from $\tilde{Y}$ to $\{y_1, \ldots, y_\kappa\}$ in D_6, for *any* choice of $x_1, \ldots, y_\kappa$ consistent with the premises of Theorem 1, and every choice for $\tilde{X}, \tilde{Y}$. We assume that we have

$$8\alpha > \epsilon_2 > r^{-1} \tag{9}$$

For $S \subseteq V$, let

$$\alpha(S) = \sum_{v \in S} |\{i : x_i = v\}| \qquad \text{and} \qquad \xi(S) = |S \cap \tilde{X}|.$$

It follows from a theorem of Gale [12] (see Bondy and Murty [3] Theorem 11.8) that if

$$d^+_{D_1}(S) \geq \xi(\bar{S}) - \alpha(\bar{S}), \qquad\qquad \forall S \subseteq V. \tag{10}$$

then there exists a flow in the network defined on D_1 such that exactly one unit of flow travels through each vertex in $\tilde{X}$ and the amount of flow traveling through each vertex $v \in \{x_1, \ldots, x_\kappa\}$ is $|\{i : x_i = v\}|$. In other words, (10) implies a successful run of Phase 2.

Now, if $|S| \leq n/2$ then, applying (4), (5) and (9), we have

$$d^+_{D_1}(S) \geq |S|\Phi_1 \geq |S|\beta_0 r \geq 8\alpha r|S| \geq \epsilon_2 r|S| \geq \alpha(S) - \xi(S) = \xi(\bar{S}) - \alpha(\bar{S}).$$

On the other hand, if $|S| > n/2$ then we have

$$d^+_{D_1}(S) = d^-_{D_1}(\bar{S}) \geq |\bar{S}|\Phi_1 \geq |\bar{S}|\beta_0 r \geq \epsilon_2 r|\bar{S}| \geq \xi(\bar{S}) - \alpha(\bar{S}).$$

Therefore, Phase 1 succeeds with respect to $X, \tilde{X}$. The same argument applies to $Y, \tilde{Y}$. To ensure these paths are of length $O(\log n)$ we can solve a minimum cost maximum flow problem as indicated in Kleinberg and Rubinfeld [13].

2.3.2 On the size of V_a

Lemma 2 *Throughout* GenPaths *we have*

$$|V_a| \geq (1 - \gamma_0)n$$

where

$$\gamma_0 = \frac{\beta_0 \gamma}{10}.$$

Proof: It follows from (8) that Δ_a is a $(\beta_0 r/2)$-expander throughout the execution of Phase 1. It follows from Lemma 1 that the diameter of Δ_a is always at most

$$\tau = \left\lceil 2 \log_3 n + \frac{2}{\alpha\gamma} \right\rceil. \tag{11}$$

Indeed, consider breadth first search from some $v \in V_a$. Let $L_t, t \geq 0$ be the vertices at distance t from v. Lemma 1 implies that the cardinalities of the L_t grow at a rate at least 3 until they reach size $(2\gamma\alpha/\beta_0)n$. The same will be true for breadth first search to a target vertex w. This accounts for the first term in (11). Once we have L_t reaches $(2\gamma\alpha/\beta_0)n$ we use that fact that going to the next level involves finding $\beta_0 r|L_t|/2$ "new arcs", at least until the size $n/2$ is reached. This accounts for the second term in (11).

Thus the total number of arcs in the paths that are removed from D_a is at most $\kappa\tau$. Let B be the set of vertices that are removed from Δ_a in the course of Phase 1, and let B_1 be the set of vertices in B incident with at least $\beta_0 r/4$ of the paths that are generated in D_a. We have

$$|B_1| \leq \frac{4\kappa\tau}{\beta_0 r} \leq \frac{\gamma_0 n}{2}$$

provided

$$\epsilon_1 \leq \frac{\beta_0^2\gamma}{160}. \tag{12}$$

where ϵ_1 is as in the statement of Theorem 1. Let $B_2 = B \setminus B_1$ (i.e. those vertices removed from Δ_a that lie on less than $\beta_0 r/4$ of the paths generated in D_a).

Assume for the sake of contradiction that $|B_2| > |B_1|$. Let B_3 be the *first* $|B_1|$ vertices of B_2 to join B. Note that the vertices in B_3 have a large degree to $B_1 \cup B_3$ (otherwise these vertices would remain in V_a). Applying (4) we have

$$in_{D_a}(B_1 \cup B_3) \geq \frac{\beta_0 r}{4}|B_3| = \frac{\beta_0 r}{8}|B_1 \cup B_3| > \alpha r|B_1 \cup B_3|.$$

This contradicts (1).

Therefore, $|B| = |B_1| + |B_2| \leq 2|B_1| \leq \gamma_0 n$. $\square$

2.3.3 Analysis of CONNECTBACK

Of course, the first order of business here is to show how the collection of paths $\mathcal{W}^{\mathrm{dir}}$ is generated.

Lemma 3 *Suppose that $D = (V, A)$ is an (α, β, γ)-expander and that $D' = (V', A')$ is a sub-digraph of D of expansion at least θr where $\theta > 6\alpha$. Suppose that $S \subseteq V'$ and that $|S| \geq (1 - \frac{\gamma\alpha}{\theta})n$ and let $T = V' \setminus S$. Then there exists $T^\star \subseteq T$, such that D' contains a collection of walks $\mathcal{W}^{\mathrm{dir}} = \{W_v : v \in T^\star\}$ such that*

1. *for* $\mathrm{dir} = \rightarrow / \leftarrow$, *$v$ is the start/terminal vertex of W_v for all $v \in T^\star$,*
2. *the terminal/start vertex of each W_v is in S,*
3. *each W_v is of length at most* $22 \log\log n$,
4. *no vertex of D' lies on more than* $240(\log\log n)^2$ *paths, and*

5. $|T \setminus T^\star| \le \frac{n}{(\log n)^4}$.

Proof: Assume without loss of generality that dir $=\rightarrow$. For $i = 1, 2, \dots$ let

$$T_i = \{v \in T : \; dist_{D'}(v, S) = i\},$$

and set $T_0 = S$. Since $N^+_{D'}(\cup_{k \ge i} T_k) \subseteq T_{i-1}$, it follows from Lemma 1 that we have

$$|T_{i-1}| \ge \zeta |T_i| \qquad \text{for } i \ge 1, \tag{13}$$

where $\zeta = \frac{\theta - \alpha}{\alpha} > 5$. Setting $i_0 = \lceil 11 \log\log n \rceil$ and $\hat{T} = \bigcup_{i \ge i_0} T_i$ It follows from (13) that we have

$$|\hat{T}| \le \frac{n}{2(\log n)^4}. \tag{14}$$

Fix $1 \le i < i_0$. We define a flow network $\mathcal{N}_i$. The vertex set of $\mathcal{N}_i$ is $\{s, t\} \cup \bigcup_{j=1}^{j_0} (C_j \cup C'_j)$ where C_1 and C'_1 are disjoint copies of T_i, for $2 \le j \le i$, C_j and C'_j are disjoint copies of $\bigcup_{\ell \ge i+1-j} T_\ell$ and, for $i < j \le j_0 = 2i_0$, C_j and C'_j are disjoint copies of V'. The vertices s and t will be the source and sink, respectively, for the flow we introduce to $\mathcal{N}_i$. A vertex v in V' may appear many times in the vertex set of $\mathcal{N}_i$; a copy of v in C_j is denoted v_j, and a copy of v in C'_j is denoted v'_j. For ease of notation, we let ϕ be the map that takes the vertices of $\mathcal{N}_i$ to their corresponding vertices in V'. The arc-set of $\mathcal{N}_i$ is defined as follows. There is an arc from s to each vertex of C_1. Each $v \in S$ gives rise to arcs $(v'_j, t), i < j \le j_0$. If $v'_j \in C'_j$ and $w_{j+1} \in C_{j+1}$ are such that (v, w) is an arc of D' then (v'_j, w_{j+1}) is an arc of $\mathcal{N}_i$. All arcs described so far have infinite capacity. In addition there are arcs (v_j, v'_j) of unit capacity defining a perfect matching between C_j and C'_j for $1 \le j \le j_0$.

Claim 1 *$\mathcal{N}_i$ contains an $s - t$ flow of value at least $|T_i| - \frac{n}{(\log n)^5}$.*

We first show the Lemma follows from Claim 1. The flow given by Claim 1 defines paths in D' from all but at most $\frac{n}{(\log n)^5}$ vertices of T_i to S, each of length at most j_0. No vertex of V' can be on more than j_0 paths since each visit to v uses a (v_j, v'_j) arc for some j. Repeating this construction for $i = 1, 2, \dots, i_0$ we find paths for all but a set $\tilde{T}$ of at most $i_0 \frac{n}{(\log n)^5} \le \frac{n}{2(\log n)^4}$ vertices and no vertex can be on more than $i_0 j_0$ paths. Putting $T^\star = T \setminus (\hat{T} \cup \tilde{T})$ and using (14) gives us the lemma.

It only remains to prove Claim 1. Of course, we do this via the max-flow min-cut theorem. Consider a cut $Z \dot{\cup} \bar{Z}$ of $\mathcal{N}_i$ where the vertex set Z contains s but not t. Let $A_j = Z \cap C_j$, $B_j = C_j \setminus A_j$, $A'_j = Z \cap C'_j$ and $B'_j = C'_j \setminus A'_j$ for $j = 1, 2, \dots, j_0$. Assume for the sake of contradiction that the capacity of this cut is less than $|C_1| - \frac{n}{(\log n)^5}$. It follows from this assumption that the cut contains no infinite capacity arcs and therefore

$$A_1 = C_1, \qquad \phi(A_{j+1}) \supseteq N^+_{D'}(\phi(A'_j)) \text{ for all } j, \qquad \text{and} \qquad \phi(A'_j) \cap S = \emptyset \text{ for all } j. \tag{15}$$

The capacity of the cut is

$$|B_1'| + \sum_{j=2}^{j_0} |\phi(A_j) \cap \phi(B_j')|. \tag{16}$$

The third condition of (15) implies that for all j, $|A_j'| < \frac{\gamma\alpha}{\theta}n$. It then follows from Lemma 1 and the second condition of (15) that we have $|A_{j+1}| \geq \zeta|A_j'|$. This implies that for all j we have either

$$|A_{j+1}'| \geq \frac{1}{2}|A_{j+1}| \geq \frac{\zeta}{2}|A_j'| \tag{17}$$

or

$$|\phi(A_{j+1}) \cap \phi(B_{j+1}')| \geq \frac{1}{2}|A_{j+1}| \geq \frac{\zeta}{2}|A_j'|. \tag{18}$$

Now, if $|A_1'| \leq \frac{n}{(\log n)^5}$ then $|B_1'| \geq |C_1| - \frac{n}{(\log n)^5}$, which contradicts our initial assumption. On the other hand, if $|A_1'| > \frac{n}{(\log n)^5}$ then, since $(\frac{\zeta}{2})^{j_0-1} > (\log n)^5$, condition (17) cannot always hold. Let $j_1 \geq 1$ be the first j for which (18) holds. We have

$$|\phi(A_{j_1+1}) \cap \phi(B_{j_1+1}')| \geq (\zeta/2)^{j_1}|A_1'| \geq |A_1'|.$$

The capacity of the cut is at least $|B_1'| + |A_1'| = |C_1|$. This is a contradiction. $\square$

To get the collection of paths needed for ConnectBack we apply Lemma 3 with $D' = \Delta_b, V' = V_b, S = V_a \cap V_b$. So, for example, we have $S_{CB}^{\text{dir}} = T^\star$. It remains to show that **whp** we have $|L| \leq \frac{n}{(\log n)^4}$.

Note that Lemma 3 can only be applied if $V_a \cap V_b$ is large. However, by applying the proof of Lemma 2 and the fact that the paths generated by ConnectBack are short (length at most $22\log\log n$) we see that $|V_b| = n - O(n\log\log n/\log n))$ throughout and this is sufficient (furthermore, the assumption that $\theta > 6\alpha$ in Lemma 3 is justified since it follows from this observation that the expansion of Δ_b is always at least $3\beta_0 r/4$). However, we shall see that V_b is **whp** larger than this. This is where we use the fact that the paths in $\mathcal{W}^{\text{dir}}$ are spread out (i.e. the fact that there are at most $240(\log\log n)^2$ paths in $\mathcal{W}^{\text{dir}}$ through any one vertex).

It follows from this fact that the probability that that an arbitrary vertex w is on the path $W_i^{CB\text{dir}}$ at most $240(\log\log n)^2/n$. It follows that we have

$$\begin{aligned}
\mathbf{Pr}\left(\left|\left\{i : w \in W_i^{CB\rightarrow}\right\}\right| + \left|\left\{i : w \in W_i^{CB\leftarrow}\right\}\right| \geq 20\right) \\
&\leq \mathbf{Pr}(B(2\kappa, 240(\log\log n)^2/n) \geq 20) \\
&\leq \binom{2\kappa}{20}\left(\frac{240(\log\log n)^2}{n}\right)^{20} \\
&= o((\log n)^{-19}).
\end{aligned} \tag{19}$$

Let B be the set of vertices which are removed from Δ_b by applications of REMOVE. Let X_1 be the set of vertices in B that are on at least 20 of the paths $W_i^{CB\to}, W_i^{CB\leftarrow}$. Note that X_1 contains the set B_1 introduced in the proof of Lemma 2 (B_1 is the collection of vertices taken out of V_b by REMOVE that are on many of the paths). It follows from (19) and Markov's inequality that **whp** we have $|X_1| \leq n/(\log n)^{18}$. Now, B (which contains $V \setminus V_b$ at every step) consists of B_1 together with extra vertices deleted by REMOVE. In total this will be at most $2|B_1|$ vertices removed by the argument of Lemma 2, following (12). Thus, $|B| \leq 2|B_1| \leq 2|X_1|$. Therefore

$$|B| \leq \frac{n}{(\log n)^{18}} \tag{20}$$

whp.

Now, a failure (i.e. the index i joining the set L) can occur in one of two ways. On one hand we have a failure if either v_i or u_i does not lie in $V_a \cup V_b$, and on the other hand a failure results when $u_i \in V_b \setminus (V_a \cup S_{CB}^{\to})$ or $v_i \in V_b \setminus (V_a \cup S_{CB}^{\leftarrow})$. It follows from (20) that the total number of failures of the first type is **whp** at most $\frac{n}{(\log n)^{18}}$. For failures of the second type we note that $v_1, \dots, v_\kappa$ form a random sequence of size $o(n)$. The probability that a particular vertex gives a failure of the second kind is at most

$$\frac{|S_{CB}^{\text{dir}}|}{n} = O\left(\frac{1}{(\log n)^4}\right).$$

Applying the Chernoff bound for the tails of the binomial, we see that, **whp** the total number of failures of the second kind is $O(n/(\log n)^4)$.

Remark 4 *Suppose D has the following vertex expansion property for small sets: $S \subseteq V$, $|S| \leq \frac{\gamma}{r}n$ implies that $|N_D^\star(S)| \geq (1-\alpha)r|S|$. The algorithm of Theorem 4 can be modified to split D so that each subgraph D_i and small S satisfies $|N_{D_i}^\star(S)| \geq \frac{1-30\alpha}{13}r|S|$. Then the shortest paths in Δ_a will be of length $O(\log_r n)$ and the claim in Remark 2 will follow.*

2.4 Phase 2.

The set of pairs $\{(\tilde{x}_i, \tilde{y}_i) : i \in L\}$ have not yet been connected by paths. We have seen that **whp** the number of such pairs, $|L|$, is at most $O(n/(\log n)^4)$. These pairs are dealt with by the algorithm described below which uses digraphs D_7–D_{13}.

The heart of the algorithm is a randomized method (based on a multicommodity flow result of Leighton and Rao [16]) for connecting pairs of vertices with arc disjoint paths which works **whp** when the collection of pairs is generated uniformly at random. So, as in Phase 1, some preliminary steps must be taken in order to reduce the problem of connecting an arbitrary set of pairs of vertices with arc-disjoint paths to the problem of connecting a random collection of pairs with arc-disjoint paths. We proceed directly to a description of the algorithm. Let $m = |L|$ and $\lambda = \lceil \log n \rceil$.

We begin by 'amplifying' each start vertex $\tilde{x}_i, i \in L$ and each end vertex $\tilde{y}_i, i \in L$ to a collection of λ vertices. This process occurs in steps **(a)** and **(b)**.

(a) In this step, we choose a collection of vertices $w_j, 1 \le j \le 2m$, and a collection of sets of vertices $W_j, 1 \le j \le 2m$, such that for $1 \le j \le 2m$,

(i) $w_j \in W_j$,

(ii) $|W_j| = \lambda$,

(iii) The sets W_j are pairwise disjoint, and

(iv) D_9 contains an arborescence with vertex set W_j and root w_j. For $1 \le j \le m$ this arborescence is directed away from the root and for $m+1 \le j \le 2m$ this arborescence is directed toward the root.

Following [17], we find these arborescences by partitioning large arborescences of D_9. We begin with a rooted spanning arborescence T with the property that all arcs are directed away from the root. We generate $W_1, \dots, W_m$ greedily from D_9, removing an arborescence from T once it is used. Of course, this process will divide T into a number of components. However, since the maximum degree of D_9 is r, the number of components produced in this process is at most $r\lambda m = O\left(\frac{nr}{(\log n)^3}\right)$. Since any tree having at least λ vertices contains a subtree having exactly λ vertices, we will always be able to find the needed arborescences. We then apply REMOVE to D_9 less the vertex set $\cup_{i=1}^m W_i$ to produce an expander D_9'. It follows from the proof of Lemma 2 that D_9' has $n - o(n)$ vertices. We repeat the process described above (this time using D_9' and an arborescense directed towards the roots) to produce $W_{m+1}, \dots, W_{2m}$.

(b) Let $S_X = \{\tilde{x}_i : i \in L\}$ and $S_Y = \{\tilde{y}_i : i \in L\}$ denote the sets of vertices that need to be joined. Use a network flow algorithm (analogous to what is given in the Initialization step of Phase 1) in D_7 to connect in an arbitrary manner the vertices of S_X to to $W_X = \{w_1, \dots, w_m\}$ by m arc disjoint paths. Using the same network flow algorithm in D_8, connect in an arbitrary manner the vertices of $W_Y = \{w_{m+1}, \dots, w_{2m}\}$ to S_Y by m arc disjoint paths. The expansion properties of D_7 and D_8 ensure that such paths always exist (as we saw in the Initialization step of Phase 1).

Let $\hat{x}_k$ (resp. $\hat{y}_k$) denote the vertex in W_X that was connected to the end-point $\tilde{x}_k$ (resp. $\tilde{y}_k$). Our problem is now to find arc-disjoint paths joining $\hat{x}_k$ to $\hat{y}_k$ for $1 \le k \le m$. If w_t has been renamed as $\hat{x}_k$ (resp. $\hat{y}_k$) then rename the elements of W_t as $\hat{x}_{k,\ell}$, (resp. $\hat{y}_{k,\ell}$) $1 \le \ell \le \lambda$.

(c) Choose $\xi_j, 1 \le j \le \lambda m$, and $\eta_j, 1 \le j \le \lambda m$, uniformly at random from V without replacement. Using a network flow algorithm (as in (b)) connect $\{\hat{x}_{k,\ell} : 1 \le k \le m, 1 \le \ell \le \lambda\}$ to $\{\xi_j : 1 \le j \le \lambda m\}$ by arc-disjoint paths in D_{10}. Similarly, connect $\{\eta_j : 1 \le j \le \lambda m\}$ to $\{\hat{y}_{k,\ell} : 1 \le k \le m, 1 \le \ell \le \lambda\}$ by arc-disjoint paths in D_{13}. Rename the other endpoint of the path starting at $\hat{x}_{k,\ell}$ (resp. ending at $\hat{y}_{k,\ell}$) as $x^*_{k,\ell}$ (resp. $y^*_{k,\ell}$).

(d) Choose $z^*_{k,\ell}, 1 \le k \le m, 1 \le \ell \le \lambda$ uniformly at random from V with replacement. (We sample without replacement in (c) to ensure that all vertices have demands 0 or 1 in the flow algorithm. Here it is convenient to sample with replacement so that these choices are independent.) Now, it is important to note that the pairs $x^*_{k,\ell}, z^*_{k,\ell}$ and the pairs $z^*_{k,\ell}, y^*_{k,\ell}$ are (when viewed separately) perfectly random. We are using the same 'trick' that we used in Phase 1 for replacing the problem of connecting arbitrary pairs to the problem of connecting random pairs.

The paths between pairs of the form $x^*_{k,\ell}, z^*_{k,\ell}$ and between pairs of the form $z^*_{k,\ell}, y^*_{k,\ell}$ are generated at random. Using the multicommodity flow algorithm of Leighton and Rao [16], find a collection of paths $P_{u,v;\theta}, u \neq v \in V, 1 \leq \theta \leq \nu_{u,v}$ where each $P_{u,v;\theta}$ is a path in D_{11} from u to v. These are the flow paths given by Theorem 18 in [16] ($\nu_{u,v}$ is simply the number of flow paths we have for the pairs u, v). Let $\mathcal{P}_{u,v} = \{P_{u,v;\theta} : 1 \leq \theta \leq \nu_{u,v}\}$. For each $P_{u,v;\theta}$ we will have a flow value $f_{u,v;\theta} > 0$ and we let $F_{u,v} = \sum_{\theta=1}^{\nu_{u,v}} f_{u,v;\theta}$. Theorem 18 promises:

(P1) For all arcs e of D_{11},
$$\sum_{(u,v,\theta):\, e \in P_{u,v;\theta}} f_{u,v;\theta} \leq 1.$$

(P2)
$$F_{u,v} \geq \frac{c_2}{n \log n}$$
for some absolute constant $c_2 > 0$.

(P3) The length of each path $P_{u,v;\theta}$ is at most $\lambda_1 = c_1 \log_r n$ for some absolute constant $c_1 > 0$.

For $u, v \in V$ let $P_{u,v}$ be the probability distribution over $\mathcal{P}_{u,v}$ where $P_{u,v}(P_{u,v;\theta}) = f(u,v,\theta)/F(u,v)$. Then for each k, ℓ choose $W'_{k,\ell}$ randomly from $\mathcal{P}_{x^*_{k,\ell}, z^*_{k,\ell}}$ using the distribution $P_{x^*_{k,\ell}, z^*_{k,\ell}}$ to select the path.

Let B'_k denote the *bundle* of paths $\{W'_{k,\ell}, 1 \leq \ell \leq \lambda\}$. Carry out the same construction in D_{12} and construct a bundle of paths $B''_k = \{W''_{k,\ell}, 1 \leq \ell \leq \lambda\}$ where $W''_{k,\ell}$ is a path from $z^*_{k,\ell}$ to $y^*_{k,\ell}$.

Let $\pi_0 = \max_e \mathbf{Pr}(e \in P)$ where P is a path chosen by (i) randomly choosing end-points u, v and then (ii) choosing $P \in \mathcal{P}_{u,v}$ according to the distribution $P_{u,v}$. We have
$$\pi_0 = \max_e \sum_{P_{u,v;\theta} \ni e} \frac{1}{n^2} \cdot \frac{f(u,v,\theta)}{F(u,v)} \leq \frac{1}{n^2} \frac{n \log n}{c_2} = \frac{\log n}{c_2 n}.$$

We say that $W'_{k,\ell}$ is *bad* if there exists $k' \neq k$ such that $W'_{k,\ell}$ shares an arc with a walk in a bundle $B'_{k'}$.

Now, suppose the bundles in the set $\{B_j : j \neq k\}$ are *fixed*. The collection of paths involved in these bundles gives at most $m\lambda\lambda_1$ arcs. Thus, the probability that $W_{k,l}$ is bad, conditioning on what happens outside the bundle B_k, is at most
$$\pi_0 m\lambda\lambda_1 = O\left(\frac{1}{\log n}\right).$$

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

$$x_k - \tilde{x}_k - \hat{x}_k - \hat{x}_{k,\ell} - x^*_{k,\ell} - z^*_{k,\ell} - y^*_{k,\ell} - \hat{y}_{k,\ell} - \hat{y}_k - \tilde{y}_k - y_k,$$

which is arc-disjoint from all other such walks.

The probability that index k is bad is at most

$$2\,\mathbf{Pr}(B(\lambda, O((\log n)^{-1}) \geq \lambda/3) = O(n^{-2}).$$

So with probability 1-o(1) there are no bad indices.

□

References

[1] N. Alon and J.H. Spencer, *The Probabilistic Method,* Wiley, 1992.

[2] J. Bang-jensen and G. Gutin, Digraphs: Theory, Algorithms and Applications, Springer-Verlag, London 2001.

[3] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North-Holland 1976.

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

[5] A. Z. Broder, A. M. Frieze, and E. Upfal, *Existence and construction of edge low congestion paths on expander graphs*, Random Structures and Algorithms 14 (1999) 87-109.

[6] S. Fortune, J.E. Hopcroft and J. Wyllie, *The directed subgraph homeomorphism problem*, Theoretical Computer Science 10 (1980) 111-121.

[7] A. Frank, *Disjoint paths in rectilinear grids*, Combinatorica 2, (1982) 361-371.

[8] A.M. Frieze, *Disjoint Paths in Expander Graphs via Random Walks: a Short Survey*, Proceedings of Random '98, Lecture Notes in Computer Science 1518 (1998) Springer, 1-14.

[9] A.M. Frieze, *Edge disjoint paths in expander graphs*, *SIAM Journal on Computing* 30 (2001) 1790-1801.

[10] A.M.Frieze and M.Molloy, *Splitting an expander graph*, Journal of Algorithms 33 (1999) 166-172.

[11] A.M. Frieze and L. Zhao, *Edge disjoint paths in random regular graphs*, Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (1999) 291-299.

[12] D. Gale, *A theorem on flows in networks*, Pacific Journal of Mathematics 7 (1957) 1073-1082.

[13] 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.

[14] J. Kleinberg and E. Tardos, *Approximations for the disjoint paths problem in high diameter planar networks*, Proceedings of the 27'th Annual ACM Symposium on Theory of Computing, (1995) 26-35.

[15] D.E. Knuth, The art of computer programming, Volume 1, Fundamental Algorithms, Addison-Wesley, 1968.

[16] T. Leighton and S. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, Journal of the Association for Computing Machinery 46 (1999) 787-832.

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

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

[19] T. Leighton, S. Rao and A. Srinivasan, *New algorithmic aspects of the local lemma with applications to partitioning and routing*, Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (1999) 643-652..

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

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

[22] N. Robertson and P. D. Seymour, *Graph minors-XIII: The disjoint paths problem*, *Journal of Combinatorial Theory Series B* 63 (1995) 65-110.

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

[24] D. Wagner and K. Weihe, *A linear time algorithm for edge-disjoint paths in planar graphs*, Proceedings of the First European Symposium on Algorithms (ESA '93) Lecture Notes in Computer Science 726, Springer-Verlag (1992) 384-395.

[25] X. Zhou, S. Tamura and T. Nishizeki, *Finding edge-disjoint paths in partial k-trees*, Algorithmica 26 (2000) 3-30.

Appendix

A Splitting an expander digraph

We prove two results on splitting D into $D_1 \cup \cdots \cup D_k$ where $D_i = (V, A_i)$. One is non-constructive and shows what might be achieved. The second is constructive and uses the first. The split produced by the second is not as good as indicated by the first result. We use a subscript i to denote graph-theoretic constructs related to D_i. Thus $d_i^+(v)$ is the out-degree of v in D_i. Left unsubscripted, such things refer to D. Thus $d^-(v) = r$.

In Section B we prove

Theorem 3 *Let $k \geq 2$ be a positive integer and let $\epsilon > 0$ be a small positive real number. Suppose that the r-regular digraph $D = (V, A)$ has edge expansion Φ and that we have*

$$\frac{r}{\ln r} \geq 7k\epsilon^{-2} \text{ and } \Phi \geq 5\epsilon^{-2} k \ln 2er.$$

Then there exists a partition $A = A_1 \cup A_2 \cup \cdots \cup A_k$ such that for $1 \leq i \leq k$

$$\Phi_i \geq (1-\epsilon)\frac{\Phi}{k}, \text{ and}$$
$$(1-\epsilon)\frac{r}{k} \leq \delta^*(D_i) \leq \Delta^*(D_i) \leq (1+\epsilon)\frac{r}{k}.$$

We then use this in Section C in the proof of

Theorem 4 *Suppose that the conditions of Theorem 3 hold, and suppose further that D is an (α, β, γ)-expander. Then there is a randomised polynomial time algorithm (running time $O(n^2 \ln n \ln \delta^{-1})$) which with probability at least $1 - \delta$ constructs $A_1, A_2, \ldots, A_k$ such that*

$$\Phi_i \geq (1-\epsilon)\frac{\Phi}{k} - (\alpha + \epsilon)\, r,$$

for $i = 1, 2, \ldots, k$.

Note that this theorem is only useful if $\Phi \geq cr$ for some c satisfying $c \gg \alpha$. For random r-regular digraphs we can take γ to be a small constant and $\alpha = O(\gamma + \frac{1}{\sqrt{r}})$. Also note that there is not enough time to verify that the algorithm succeeds. Instead, we assume it has and repeat the split if we fail to find the required paths.

B Existence Result

We prove Theorem 3. We will use the general version of the Lovász Local Lemma. For each $a \in A$ we randomly choose an integer $i \in [k]$ and then place a in A_i. We must show that there is a positive probability of choosing a partition which satisfies the conditions of the theorem.

We begin with some definitions and preliminary observations. Let $G = (V, E)$ be the $2r$-regular (multi-)graph obtained by ignoring orientation in D. If $S \subseteq V$ then $G[S]$ is the subgraph of G induced by S. We say that S is *connected* if $G[S]$ is.

Claim 2 *For $v \in V$ there are at most $(2er)^{s-1}$ sets S such that (i) $v \in S$, (ii) $|S| = s$ and (iii) S is connected.*

Proof of Claim 2 The number of such sets is bounded by the number of distinct s-vertex trees which are rooted at v. This in turn is bounded by the number of distinct $2r$-ary rooted trees with s vertices. This is equal to $\binom{2rs}{s}/((2r-1)s+1)$, see Knuth [15]. $\square$

In both this section and the sequel we will use the following Chernoff bounds for the tails of the binomial distribution $B(n, p)$:

$$\mathbf{Pr}(B(n,p) \geq (1+\epsilon)np) \leq e^{-\epsilon^2 np/3} \tag{21}$$

$$\mathbf{Pr}(B(n,p) \leq (1-\epsilon)np) \leq e^{-\epsilon^2 np/2} \tag{22}$$

where $0 \leq \epsilon \leq 1$.

For our application of the Lovász Local Lemma, we define the following *bad* events:

(a) For $v \in V$, $i \in [k]$ and $* \in \{+,-\}$, $A_{v,i,*} = A_{\{v\},i,*}$ is the event that

$$d_i^*(v) \notin [(1-\epsilon)r/k, (1+\epsilon)r/k].$$

(b) For $S \subseteq V$, $2 \leq |S| \leq n/2$, S connected, $i \in [k]$ and $* \in \{+,-\}$, $A_{S,i,*}$ is the event that

$$|d_i^*(S)| < (1-\epsilon)|d^*(S)|/k.$$

In showing that Φ_i is sufficiently large we can restrict our attention to S for which S is connected. Indeed, for $S \subseteq V$ let $C_1, C_2, \ldots, C_t$ be the components of $G[S]$. Then for $* \in \{+,-\}$,

$$\Phi_{S,i}^* \geq \min_{1 \leq s \leq t} \frac{d_i^*(C_s)}{|C_s|}.$$

Using the Chernoff bounds given above we obtain,

$$\mathbf{Pr}(A_{v,i,*}) \leq 2e^{-\epsilon^2 r/(3k)} \leq 2e^{-(7 \ln r)/3} < \frac{1}{r^2}$$

and

$$\mathbf{Pr}(A_{S,i,*}) \leq \exp\left\{-\frac{\epsilon^2 d^*(S)}{2k}\right\} \leq e^{-2|S| \ln r} = \frac{1}{r^{2|S|}}.$$

Now, for $S \subseteq V$, $1 \leq |S| \leq n/2$ and S connected, let

$$x_{S,i,*} = \left(\frac{2}{r^2}\right)^{|S|}.$$

We show that for $*, \# \in \{+, -\}$,

$$\mathbf{Pr}(A_{S,i,*}) < x_{S,i,*} \prod_{(S,i,*)\sim(T,j,\#)} (1 - x_{T,j,\#}), \tag{23}$$

where $(S, i, *) \sim (T, j, \#)$ denotes adjacency of $A_{S,i,*}$ and $A_{T,j,\#}$ in the dependency graph of bad events (i.e. we have $(S, i, *) \sim (T, j, \#)$ if and only if $A^*(S) \cap A^\#(T) \neq \emptyset$). The theorem then follows from the general version of the local lemma, see for example Alon and Spencer [1].

It follows from Claim 2 that if $|S| = s$ then there are at most $ks(2er)^t$ events $A_{T,j,\#}$ with $|T| = t$ such that $(S, i, *) \sim (T, j, \#)$. Thus, using $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$ we have

$$\begin{aligned} x_{S,i,*} \prod_{(S,i,*)\sim(T,j,\#)} (1 - x_{T,j,\#}) &\geq \left(\frac{2}{r^2}\right)^s \prod_{t\geq 1} \left(1 - \left(\frac{2}{r^2}\right)^t\right)^{ks(2er)^t} \\ &\geq \left(\frac{2}{r^2}\right)^s \exp\left\{-2ks \sum_{t\geq 1} \left(\frac{4e}{r}\right)^t\right\} \\ &= \left(\frac{2}{r^2}\right)^s \exp\left\{-\frac{8kes}{r - 4e}\right\} \\ &> \frac{1}{r^{2s}}, \end{aligned}$$

since for small values of ϵ, the fact that $r/\ln r \geq 7k\epsilon^{-2}$ implies

$$r > 4e + \frac{8ke}{\ln 2}.$$

Thus (23) holds, proving the theorem. $\square$

C Splitting Algorithm

In this section, we prove Theorem 4.

Idea: We produce the split in a series of rounds that gives a series of vertex sets $V = B_1 \supseteq B_2 \supseteq \cdots \supseteq B_t$. In round i we fix the 'destination' of a fixed arc if it has at least one endpoint in B_i but has no enpoint in B_{i+1}. This is done in such a way that if $S \subseteq B_j \setminus B_{j+1}$ then $d_i^+(S), d_i^-(S)$ are large enough and further that every vertex in $B_j \setminus B_{j+1}$ has few neighbors in B_{j+1}. We will see that this latter condition accounts for the $-(\alpha + 2\epsilon)r$ term in the theorem.

Assume we have $B \subseteq V$. Initially, $B = V$. We randomly colour the arcs of D **which**

are incident with B, with k colours. Note that if $s_0 = 5k\epsilon^{-2}\Phi^{-1}\ln n$

$$\mathbf{Pr}\left(\exists S \subseteq B, i \in [k] \text{ s.t. } |S| > s_0, S \text{ is connected and } \Phi_{i,S} \le (1-\epsilon)\frac{\Phi}{k}\right)$$
$$\le 2kn\sum_{s\ge s_0}(2er)^{s-1}e^{-\epsilon^2 s\Phi/(2k)} \le 4kn(2er)^{s_0}e^{-\epsilon^2 s_0\Phi/(2k)} = O(kn^{-1/5}).$$

So, in a sense the large sets, take care of themselves. Now consider the smaller sets. Let

$$X_0 = \left\{v : \exists S \subseteq B, |S| \le s_0, S \text{ is connected}, v \in S \text{ and } i \in [k] \text{ s.t. } \Phi_{i,S} \le (1-\epsilon)\frac{\Phi}{k}\right\}.$$

X_0 can be constructed in $O(n(er)^{s_0}) = O(n^2)$ time.

$$\mathbf{E}(|X_0|) \le |B|\sum_{s=1}^{s_0}(2er)^{s-1}e^{-\epsilon^2 s\Phi/(2k)} \le \frac{|B|}{2er}$$

since $\Phi \ge 5\epsilon^{-2}k\ln 2er$.

Therefore by Markov's Inequality,

$$\mathbf{Pr}\left(|X_0| \ge \frac{|B|}{er}\right) \le \frac{1}{2}.$$

We repeat the above colouring until we find that $|X_0| \le \frac{|B|}{er}$. Now recursively define $X_j = X_{j-1} \cup \{v_j\}$ where $d^+(v_j, X_{j-1}) \ge (\alpha+\epsilon)r$ or $d^-(v_j, X_{j-1}) \ge (\alpha+\epsilon)r$, if such a v_j exists. Here we use the strong expansion properties of an (α,β,γ)-expander: if $|S| \le \gamma n$ implies that S contains at most

$$(r|S| - d^*(S)) \le \alpha r|S|$$

arcs. Note that X_j has at least $(\alpha+\epsilon)rj$ arcs and at most $j + \frac{|B|}{er}$ vertices. Thus this process stops before j reaches $\frac{\alpha|B|}{\epsilon er}$, unless $|X_j|$ exceeds γn first. However, this latter possibility cannot happen since $|X_0| + \frac{\alpha|B|}{\epsilon er} \le (1+\frac{\alpha}{\epsilon})\frac{1}{er}|B| \le \gamma|B| \le \gamma n$ since $\gamma, \epsilon > r^{-\frac{1}{2}}, \alpha < 1$ implies $\gamma > (1+\frac{\alpha}{\epsilon})\frac{1}{er}$.

So if X denotes X_j when v_{j+1} cannot be found, then

$$|X| \le \gamma|B|.$$

We will repeat the construction with B replaced by X. Let $V = B_1 \supseteq B_2 \supseteq \cdots \supseteq B_t$ be the sequence of sets constructed. B_t will be the first set of size at most $r^{-1}\ln n$. Since $\gamma < \frac{1}{2}$, we have $t \le \log_2 n$. Thus the expected number of re-colourings needed is at most $2\log_2 n$ and is $\le 3\log_2 n$ **whp**. We can "brute force" colour the arcs incident with B_t so that every subset S of B_t satisfies $\Phi_{i,S} \ge (1-\epsilon)\frac{\Phi}{k}$. We use Theorem 3 to justify the success of this. The sequence of sets $B_1, B_2, \ldots, B_t$ satisfies

- $|B_j| \leq \gamma^j n$.
- $S \subseteq B_j \setminus B_{j+1}$ implies $\Phi_{i,S} \geq (1-\epsilon)\,\frac{\Phi}{k}$.
- $v \in B_j \setminus B_{j+1}$ implies v has at most $(\alpha+\epsilon)r$ out-neighbours and at most $(\alpha+\epsilon)r$ in-neighbours in B_{j+1}.

So if $S \subseteq V$ and $S_j = S \cap (B_j \setminus B_{j+1})$,

$$\begin{aligned} d_i^*(S) &\geq \sum_{j=1}^{t-1} (d_i^*(S_j) - d_i^*(S_j, B_{j+1})) + d_i^*(S_t) \\ &\geq \sum_{j=1}^{t-1} \left((1-\epsilon)\frac{\Phi}{k} - (\alpha+\epsilon)r \right) |S_j| + (1-\epsilon)\,\frac{\Phi}{k}|S_t| \\ &\geq \left((1-\epsilon)\frac{\Phi}{k} - (\alpha+\epsilon)\,r \right) |S|. \end{aligned}$$

□