Electronic Colloquium on Computational Complexity, Report No. 22 (2005)

Pseudorandom Walks in Biregular Graphs and the $\mathbf{RL}$ vs. $\mathbf{L}$ Problem*

PRELIMINARY VERSION

Omer Reingold† Luca Trevisan‡ Salil Vadhan§

February 4, 2005

Abstract

Motivated by Reingold's recent deterministic log-space algorithm for UNDIRECTED S-T CONNECTIVITY (ECCC TR 04-94), we revisit the general $\mathbf{RL}$ vs. $\mathbf{L}$ question, obtaining the following results.

1. We exhibit a new complete problem for $\mathbf{RL}$: S-T CONNECTIVITY restricted to directed graphs for which the random walk is promised to have polynomial mixing time.
2. Generalizing Reingold's techniques, we present a deterministic, log-space algorithm that given a directed graph G that is *biregular* (i.e., all in-degrees and out-degrees are equal) and two vertices s and t, finds a path between s and t if one exists.
3. Using the same techniques as in Item 2, we give a "pseudorandom generator" for random walks on "consistently labelled" biregular graphs. Roughly speaking, given a random seed of logarithmic length, the generator constructs, in log-space, a "short" pseudorandom walk that ends at an almost-uniformly distributed vertex when taken in any consistently labelled biregular graph.
4. We prove that if our pseudorandom generator from Item 3 could be generalized to all biregular graphs (instead of just consistently labelled ones), then our complete problem from Item 1 can be solved in log-space and hence $\mathbf{RL} = \mathbf{L}$.

*Research supported by US-Israel Binational Science Foundation Grant 2002246.

†Incumbent of the Walter and Elise Haas Career Development Chair, Department of Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. omer.reingold@weizmann.ac.il

‡U.C. Berkeley. luca@cs.berkeley.edu

§Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. Also supported by NSF grant CCR-0133096, ONR grant N00014-04-1-0478, and a Sloan Research Fellowship. salil@eecs.harvard.edu

ISSN 1433-8092

1 Introduction

The research on derandomization of space-bounded computations deals with the tradeoff between two basic resources of computations: memory (or space) and randomness. Can randomness save space in computations? Alternatively, can every randomized algorithm be derandomized with only a small increase in space? These questions received the most attention in the context of log-space computations, and with respect to the following complexity classes: $\mathbf{L}$ (the class of problems solvable in deterministic log-space), $\mathbf{RL}$, and $\mathbf{BPL}$ (the classes of problems solvable by randomized log-space algorithms making one-sided and two-sided errors respectively). It is widely believed that $\mathbf{L} = \mathbf{RL} = \mathbf{BPL}$ and proving this conjecture is the ultimate goal of this body of research.

It turns out that the derandomization of $\mathbf{RL}$ is related to determining the space complexity of one of the most basic graph problems, UNDIRECTED S-T CONNECTIVITY: Given an undirected graph and two vertices, is there a path between the vertices? (The corresponding search problem is to find such a path). The space complexity of this problem and the derandomization of space-bounded computations have been the focus of a vast body of work, and brought about some of the most beautiful results in complexity theory. The connection between the two was made by Aleliunas et. al. [AKL$^+$], who gave an $\mathbf{RL}$ algorithm for UNDIRECTED S-T CONNECTIVITY. The algorithm simply runs a random walk from the first vertex s for a polynomial number of steps, and accepts if and only if the walk visits the second vertex t. This beautifully simple algorithm is undoubtedly one of the most interesting $\mathbf{RL}$ algorithms. It casts the space complexity of UNDIRECTED S-T CONNECTIVITYas a specific example and an interesting test case for the derandomization of space-bounded computations. (In particular, if $\mathbf{RL} = \mathbf{L}$, then UNDIRECTED S-T CONNECTIVITY can be solved in deterministic log-space.) Since then progress on the general and the specific problems alternated with a fluid exchange of ideas (as demonstrated by [Sav, AKS, BNS, Nis2, Nis1, NSW, SZ, ATSWZ], to mention just a few highlights of this research). See the surveys of Saks [Sak] and Wigderson [Wig] for more on these vibrant research areas.

The starting point of our research is a recent result of Reingold [Rei] that showed that UNDIRECTED S-T CONNECTIVITYhas a deterministic log-space algorithm. On the other hand, the best deterministic space bound on $\mathbf{RL}$ in general remains $O(\log^{3/2} n)$, established by Saks and Zhou [SZ]. In this paper, we revisit the general $\mathbf{RL}$ vs. $\mathbf{L}$ question in light of Reingold's results, and obtain the following results:

1. We exhibit a new complete problem for $\mathbf{RL}$: S-T CONNECTIVITY restricted to directed graphs for which the random walk is promised to have polynomial "mixing time," as measured by a directed analogue of the spectral gap introduced by Mihail [Mih].

2. Generalizing Reingold's techniques, we present a deterministic, log-space algorithm that given a directed graph G which is *biregular* (i.e. all indegrees and outdegrees are equal) and two vertices s and t, finds a path between s and t if one exists. This involves a new analysis of the zig-zag graph product of [RVW] that generalizes to biregular graphs and the directed analogue of the spectral gap, which may be of independent interest.

3. Using the same techniques as in Item 2, we give a "pseudorandom generator" for random walks on "consistently labelled" biregular graphs. [1] Roughly speaking, given a random seed of logarithmic

[1] A D-regular biregular graph is consistently labelled if the D edges leaving each vertex are numbered from 1 to D in such a way that at each vertex, the labels of the incoming edges are all distinct. It turns out that every biregular graph has a consistent labelling, but this does not imply that a pseudorandom generator that works for consistently labelled graphs also works for arbitrarily labelled biregular graphs.

length, the generator constructs, in log-space, a "short" pseudorandom walk that ends at an almost-uniformly distributed vertex when taken in any consistently labelled biregular graph.

4. We prove that if our pseudorandom generator from Item 3 could be generalized to all biregular graphs (instead of just consistently labelled ones), then our complete problem from Item 1 can be solved in log-space and hence $\mathbf{RL} = \mathbf{L}$.

Thus, the "only" obstacle to proving $\mathbf{RL} = \mathbf{L}$ is the issue of consistent labelling. It remains to be seen how difficult this is to overcome.

1.1 Techniques

In this section, we give an overview of our techniques,

Reingold's algorithm. The algorithm of [Rei] can be seen as a logarithmic space reduction from the st-connectivity problem in general undirected graphs to the st-connectivity problem restricted to graphs such that each connected component is a regular constant-degree expander. The latter problem is easily solvable deterministically in logarithmic space by observing that a constant-degree expander has logarithmic diameter, and so one can test if two vertices are in the same connected component by enumerating all walks of logarithmic length starting at one of the vertices.

More specifically, the reduction starts from a graph G, that we may assume without loss of generality to be regular of constant degree and two vertices s and t, and it produces a new constant degree regular graph G' and vertices s' and t' such that (i) s' and t' are connected iff s and t are, and (ii) every connected component of G' is an expander.

The reduction works in phases: in each phase the size of the graph grows by a constant factor, the degree remains the same, and the *spectral gap* (the difference between largest and second largest eigenvalues of the transition matrix of the random walk on G) improves by a constant factor. Initially, in each connected component the spectral gap is at least $1/\mathrm{poly}(n)$, and so, after $O(\log n)$ phases, the spectral gap is a constant, which implies that the diameter of each connected component is logarithmic, and the size of the final graph is polynomial. The final graph can be computed in space linear in the number of phases, and so logarithmic space is sufficient to carry out such a reduction.

Each phase of the reduction consists, in turn, of two steps. In the first step, the (adjacency matrix of the) graph is raised to a constant power, an operation that improves the spectral gap in each connected component by a constant factor, at the cost of increasing the degree. In the second step, we take the *zig-zag* graph product [RVW] with a constant-size constant-degree expander. This reduces the degree back to a fixed constant, at the cost of increasing the number of vertices and making the spectral gap worse. If the parameters are carefully balanced, then, as we said, each phase leaves the degree unchanged, increases the number of vertices by a constant factor, and improves the spectral gap by a constant factor.

With an eye towards extending these techniques to all of $\mathbf{RL}$, let us phrase the overall structure of the argument more abstractly.

- Define a measure of expansion for undirected graphs. Specifically, consider the spectral gap $\gamma(G)$, the difference between largest and second largest eigenvalue of the transition matrix of the random walk on G.

- Observe that if $\gamma(G) = \Omega(1)$ (and G is of constant degree), then deciding st-connectivity in G is easy.

- Note that for every undirected graph G on n vertices, $\gamma(G) \geq 1/\mathrm{poly}(n)$.
- Introduce an operation (powering) that increases $\gamma(G)$ but increases the degree.
- Introduce an operation (zig-zag with an expander) that reduces the degree while slightly reducing the $\gamma(G)$.
- Show that the graph obtained by applying $O(\log n)$ operations as above to G can be computed in logarithmic space given G.

Generalizing to $\mathbf{RL}$. Given the above view of [Rei], we explore how each of the steps can be generalized to arbirary $\mathbf{RL}$ algorithms.

Measure of Expansion: We consider a generalization $\gamma(G)$ of the spectral gap to directed graphs, introduced by Mihail [Mih] and Fill [Fil], in the context of nonreversible Markov chains. However, in general graphs, this parameter does not have all of the nice properties utilized in the logspace algorithm for the undirected case. First, there are graphs G such that $\gamma(G)$ is exponentially small. Second, $\gamma(G)$ being constant does not imply logarithmic diameter. Instead, it only implies that there is a path of length $O(\log n)$ from s to t *provided* that t has probability mass at least $1/\mathrm{poly}(n)$ in the stationary distribution of the random walk on G. Fortunately, as discussed in the next item, we are able to show that every $\mathbf{RL}$ problem can be reduced to solving S-T CONNECTIVITY on graphs where neither of these difficulties occur.

A New Complete Promise Problem: We prove that the following promise problem, POLY-MIXING S-T CONNECTIVITY, is complete for $\mathbf{RL}$: we are given a directed graph G, two nodes s and t, and the promise either that t is unreachable from s or that $\gamma(G) \geq 1/\mathrm{poly}(n)$ and s and t have probability mass at least $1/\mathrm{poly}(n)$ in the stationary distribution of the random walk that starts at s.

Powering: We note that, just as in the undirected case, powering improves $\gamma(G)$ and does not change the stationary distribution (at the price of increasing the degree).

The Zig-Zag Product. By the above description, the main remaining challenge would be to show that there is an operation that reduces the degree while not affecting by much the stationary distribution and affecting only slightly the expansion γ. This turns out to be rather tricky, and in particular, for the zig-zag product we are aware of examples where it does not work. Interestingly, the part that seems the hardest (and fails miserably in the negative examples), is maintaining the nonnegligible probability of t under the stationary distribution.

Solving the Biregular Case. One interesting case where we are able to show that the zig-zag product with an expander satisfies the desired properties is when we start from a *biregular* directed graph. We note that for biregular graphs, the stationary distribution is uniform and therefore it is easier to maintain its uniformity (as long as we make sure to maintain biregularity). The analysis of expansion in the biregular case, however, requires a more general approach than the one taken in [RVW] (which exploits linear-algebraic and geometric consequences of the symmetric adjacency matrices associated with undirected graphs). At the same time, this analysis gives a much simpler proof for the bound on the zig-zag product needed for [Rei].

We are able to generalize the algorithm of [Rei] to a log-space deterministic algorithm that finds paths in general biregular graphs (note that there is no promise in this problem).[2] In addition, this algorithm implies a log-space constructible *universal traversal sequence* for biregular graphs with *consistent labelling*. A **universal traversal sequence** for a family of labelled graphs is a sequence of edge labels that leads a walk from any start vertex of any graph in the family through all the other vertices of the graph. A labelling of the edges of a graph is **consistent** if all the edge-labels that lead to any particular vertex are distinct. We note that this is the least restrictive condition under which such universal traversal sequences are known *even for expander graphs* [HW]. Furthermore, even when restricted to the undirected case, this condition on the labelling is less restrictive than the one in the universal traversal sequences that are directly implied by the algorithm of [Rei].

Finally, we give a generator with logarithmic seed length that produces in log-space a "pseudorandom walk" for consistently labelled biregular graphs. This means that, from any start vertex, the walk produced by the generator on a random seed converges to the (uniform) stationary distribution. A useful property is that the generator produces shorter pseudorandom walks for graphs with better expansion. More precisely, the length of the walk depends polynomially on $1/\gamma(G)$, but only *logarithmically* on the number of vertices of the graph. This matches, up to a polynomial, the length of a truly random walk achieving comparable mixing.

Derandomizing $\mathbf{RL}$ Reduces to the Biregular Case. As mentioned above, applying the techniques of [Rei] to tackle the general $\mathbf{RL}$ case is challenging. Particularly, it is not clear what is an appropriate analogue to the zig-zag product in the general case. That is, we do not have a candidate operation that could reduce the degree while nearly preserving the expansion parameter and the weight of t under the stationary distribution. We therefore propose an alternative approach to reducing the general case to the biregular case. This seems to be a non-trivial task (unlike st-connectivity in undirected graphs which easily reduces to st-connectivity in regular graphs). Still we provide a method of solving the new complete problem for $\mathbf{RL}$ using a generator with logarithmic seed length that produces in log-space a "pseudorandom walk" for biregular graphs. As mentioned above, this paper obtains a pseudorandom generator for walks on biregular graphs. Furthermore, most of the properties of this generator are more than sufficient for solving $\mathbf{RL}$ using our reduction. The only deficiency of that generator is that it works under the assumption that the edge labels are consistent rather than working for biregular graphs with *arbitrary labelling*.

On Alternative Measures of Expansion. It is natural to attempt the general framework of derandomization studied here with a different measure of expansion. We also consider the combinatorial measure of *edge expansion*. We show that edge expansion is preserved and degree is reduced, by taking a *replacement product* with an expander graph. We show, however, that edge expansion is not necessarily improved by powering in directed graphs, and it is not clear that there is any other "local" operation that increases edge expansion.

1.2 Perspective

We feel that the combination of our two results regarding biregular graphs sheds promising light on the $\mathbf{L}$ vs. $\mathbf{RL}$ problem: First, derandomizing general $\mathbf{RL}$ computations reduces to derandomizing (via pseudorandom generators) the random walk on a biregular graph. This seems a closely related task to derandomizing

[2] We note that the decision problem in biregular graphs rather easily reduces to the undirected case, but this does not seem to be the case for the search problem.

the random walk on an undirected graph. Secondly, the difficulty in derandomizing the random walk is in the labelling of the graph. A possible explanation for the importance of the labelling is that for any fixed sequence of labels, the corresponding walk on a graph with consistent labels cannot lose entropy (the distribution of the final vertex has as much entropy as the distribution of the start vertex). On the other hand, without the assumption on the labelling, entropy losses may occur. It is therefore harder to argue that progress made in a pseudorandom walk (ie increase in entropy) will not be lost later in a walk.

1.3 Organization

We begin by defining notions of expansions for directed graphs and giving other technical preliminaries in Section 2. We present in Section 3 our new $\mathbf{RL}$-complete promise problem. The operations of powering, replacement product and zig-zag graph product are defined for directed graphs in Section 4, and the effect of powering and zig-zag product on biregular directed graphs is analysed in Section 5, leading to our algorithm for finding paths in biregular graphs, to our construction of universal transversal sequences for biregular consistently labeled graphs, and our pseudorandom walk generator for biregular consistently labeled graphs. In Section 6 we prove that a pseudorandom walk generator for general biregular graphs would imply $\mathbf{L} = \mathbf{RL}$. We conclude with a discussion of other measures of expansion in Section 7.

2 Preliminaries

2.1 Graphs and Markov Chains

In this paper, we allow **graphs** to be directed, have multiple edges, and have self-loops. A graph is **out-regular** (resp., **in-regular**) if every vertex has the same number D of edges leaving it; D is called the **out-degree** (resp., **in-degree**). A graph is **biregular** if it is both out-regular and in-regular.

Given a graph G on N vertices, we consider the random walk on G described by the transition matrix M_G whose (v,u)'th entry equals the number of edges from u to v, divided by the outdegree of v.[3]

We are interested in the rate at which random walks on G converge to a **stationary distribution** (i.e. a probability mass function $\pi : [N] \to [0,1]$ such that $M_G\pi = \pi$). In the case that G is undirected (i.e. the Markov chain is time-reversible), it is well-known that the rate of convergence is characterized by the second largest eigenvalue $\lambda = \lambda(G)$ of the symmetric matrix M. (The **spectral gap** discussed in the introduction is simply $\gamma(G) = 1 - \lambda(G)$.)

Following Mihail [Mih] and Fill [Fil], we consider a generalization of the parameter λ to the directed case. For a probability mass function $\pi : [N] \to [0,1]$ on vertices, we define a normalized inner product on $\mathbb{R}^N$ by:

$$\langle x, y\rangle_\pi \stackrel{\text{def}}{=} \sum_{v\in[N]} \frac{x(v)\cdot y(v)}{\pi(v)},$$

and a norm $\|x\|_\pi \stackrel{\text{def}}{=} \sqrt{\langle x,x\rangle_\pi}$. Note that this normalization makes π itself a unit vector (i.e. $\|\pi\|_\pi = 1$), and also implies that x is orthogonal to π iff $\sum_v x(v) = 0$.

Definition 2.1 *Let G be a directed graph, s a vertex from which all vertices of G are reachable, and π a probability function on the vertices of G. We say that G has* **spectral expansion** *λ with respect to s and π if*

[3] Often the transition matrix is defined to be the transpose of our definition. Our choice means taking a random walk corresponds to *left*-multiplication by M_G.

1. *π is a stationary distribution of G, and*

2. *$\lambda_{s,\pi}(G) \leq \lambda$, where*

$$\lambda_{s,\pi}(G) = \max_{x \in \mathbb{R}^N : \langle x, \pi \rangle_\pi = 0} \frac{\|M_G x\|_\pi}{\|x\|_\pi},$$

If not all vertices in G are reachable from s, then we define $\lambda_{s,\pi}(G) = \lambda_{s,\pi}(G')$, where G' is the induced subgraph consisting of all vertices reachable from s. (In particular, we require that the support of π is contained in G'.) We define $\lambda_s(G) = \min_\pi \lambda_{s,\pi}(G)$.

The reason that the above spectral expansion λ captures the convergence rate of the Markov chain is that if α is our initial distribution (say concentrated at the start vertex s), then $\alpha - \pi$ is orthogonal to π (with respect to the normalized inner product). Thus,

$$\|M^t \alpha - \pi\|_\pi = \|M^t(\alpha - \pi)\|_\pi \leq \lambda_{s,\pi}(G)^t \cdot \|\alpha - \pi\|_\pi,$$

so the (normalized) ℓ_2-distance to π decreases exponentially fast, with a base of $\lambda_{s,\pi}(G)$. In particular, if $\lambda_{s,\pi}(G) < 1$, then π is the unique stationary distribution for which this holds, and we denote it by π_s.

Just as in the undirected case, the spectral expansion can be bounded in terms of the sizes of cuts in the underlying graph.

Definition 2.2 *Let G be a directed graph on N vertices and π a stationary distribution of the random walk on G. For a set of vertices A such that $\pi(A) > 0$, we define $\Phi_\pi(A) = \Pr[X' \notin A | X \in A]$, where X is a vertex chosen according to the stationary distribution π and X' is obtained by taking a random step in G from X. The* **conductance** *of G with respect to π is defined to be*

$$h_\pi(G) \stackrel{\text{def}}{=} \min_{A : 0 < \pi(A) \leq 1/2} \Phi_\pi(A).$$

Lemma 2.3 ([SJ, Mih, Fil]) *Let G be a directed graph on N vertices in which at least half of the edges leaving every vertex in G are self-loops (i.e. the random walk on G is "strongly aperiodic"), and let π be a stationary distribution of G. Then for every vertex s in the support of π, $\lambda_{s,\pi}(G) \leq 1 - h_\pi(G)^2/2$.*

When the stationary distribution π is uniform on the vertices of G, then the conductance defined above coincides exactly with the "edge expansion" of G, defined below. [4]

Definition 2.4 *Let $G = (V, E)$ be a directed graph in which every vertex has outdegree D. Then the* **edge expansion** *of G is defined to be*

$$\varepsilon(G) = \min_A \frac{E(A, \overline{A})}{D \cdot \min\{|A|, |\overline{A}|\}},$$

where the minimum is taken over sets of vertices A and $E(A, \overline{A})$ is the set of edges (u, v) where $u \in A$ and $v \notin A$.

[4] To see that $\varepsilon(G) = h_\pi(G)$ when π is the uniform distribution, note that the fact that the stationary distribution is uniform implies that G is biregular, which in turn implies that $E(A, \overline{A}) = E(\overline{A}, A)$.

2.2 Complexity Classes

$\mathbf{L}$, $\mathbf{RL}$, $\mathbf{NL}$, $\mathbf{BPL}$ are the standard logspace complexity classes. $\mathbf{prL}$, $\mathbf{prRL}$, $\mathbf{prBPL}$ are the promise-problem versions of these classes. Now we define search versions of logspace classes. An **search problem** $\Pi = (R, L)$ is specified by a relation $R \subseteq \Sigma^* \times \Sigma^*$ and a "promise language" $L \subseteq \{x : R(x) \neq \emptyset\}$, where $R(x) = \{y : (x, y) \in R\}$. The associated computational problem is: given $x \in L$, output any string $y \in R(x)$. Π is in $\mathbf{searchNL}$ if R is polynomially balanced and $(x, y) \in R$ can be decided in logspace given two-way access to x and one-way access to y. Π is in $\mathbf{searchL}$ if there is a logspace algorithm A (with one-way access to its output) such that for every $x \in L$, $A(x) \in R(x)$. (It can be verified that the promise language L is in $\mathbf{L}$.) R is in $\mathbf{searchRL}$ if it is in $\mathbf{searchNL}$ and there is a randomized logspace algorithm A (with one-way access to its output) that outputs a sequence of of strings $A(x) = (y_1, \ldots, y_t)$ such that for every $x \in L$, $\Pr[A(x) \cap R(x) \neq \emptyset] \geq 1/2$. Equivalently, there is a randomized logspace algorithm A outputting a single string such that $\Pr[A(x) \in R(x)] \geq 1/\mathrm{poly}(n)$.

All reductions in this paper are logspace reductions. For search problems, the definition is a bit subtle. A search problem $\Pi = (R, L)$ reduces to search problem $\Pi' = (R', L')$ if there are functions $f(x)$ and $g(x, z)$ such that if $x \in L$ then $f(x) \in L'$, and if $z \in R'(f(x))$, then the *sequence* $g(x, z) = (y_1, \ldots, y_\ell)$ contains at least one element of $R(x)$. f should be computable in logspace with two-way access to x, and g should be computable in logspace with two-way access to x and one-way access to z. It can be verified that if $\Pi \in \mathbf{searchNL}$ reduces to Π', then $\Pi' \in \mathbf{searchL} \Rightarrow \Pi \in \mathbf{searchL}$.

Proposition 2.5 $\mathbf{prBPL} = \mathbf{prL}$ *implies* $\mathbf{searchRL} = \mathbf{searchL}$.

3 A New Complete Problem for RL

S-T CONNECTIVITY and its search version, FIND PATH, both defined below, are two of the most basic problems in computer science.

S-T CONNECTIVITY**:**

- **Input:** (G, s, t), where $G = (V, E)$ is a directed graph, $s, t \in V$
- **YES instances:** There is a path from s to t in G.
- **NO instances:** There is no path from s to t in G.

FIND PATH**:**

- **Input:** (G, s, t), where $G = (V, E)$ is a directed graph, $s, t \in V$, and $k \in \mathbb{N}$
- **Promise:** There is a path from s to t in G.
- **Output:** A path from s to t in G.

It is well-known that S-T CONNECTIVITY is complete for $\mathbf{NL}$, and the same argument shows that FIND PATH is complete for $\mathbf{searchNL}$. Here we are interested in the complexity of restrictions of these problems. The recent result of Reingold [Rei] shows that their restrictions to *undirected* graphs, UNDIRECTED S-T CONNECTIVITY and UNDIRECTED FIND PATH, are in $\mathbf{L}$ and $\mathbf{searchL}$, respectively.

It was known (see [Sak]) that a certain restriction of S-T CONNECTIVITY was complete for $\mathbf{prRL}$, specifically one where we look at the probability that a random walk of a particular length goes from s to t:

SHORT-WALK S-T CONNECTIVITY**:**

- **Input:** $(G, s, t, 1^k)$, where $G = (V, E)$ is a directed graph, $s, t \in V$
- **YES instances:** A random walk of length k started from s ends at t with probability at least $1/2$.
- **NO instances:** There is no path from s to t in G.

However, this problem does not seem to capture the properties of UNDIRECTED S-T CONNECTIVITY used in Reingold's algorithm [Rei]. As described in the introduction, his algorithm uses relies on a measure of expansion, specifically the spectral gap, which refers to the *long-term* behavior of random walks in G (as opposed to walks of a particular length k). We give a complete problem that seems much closer, specifically by restricting to graphs of polynomial mixing time (as measured by $\lambda_s(G)$).

POLY-MIXING S-T CONNECTIVITY**:**

- **Input:** $(G, s, t, 1^k)$, where $G = (V, E)$ is a out-regular directed graph, $s, t \in V$, and $k \in \mathbb{N}$
- **YES instances:** $\lambda_s(G) \leq 1 - 1/k$, and $\pi_s(s), \pi_s(t) \geq 1/k$.
- **NO instances:** There is no path from s to t in G.

POLY-MIXING FIND PATH**:**

- **Input:** $(G, s, t, 1^k)$, where $G = (V, E)$ is a out-regular directed graph, $s, t \in V$, and $k \in \mathbb{N}$
- **Promise:** $\lambda_s(G) \leq 1 - 1/k$, and $\pi_s(s), \pi_s(t) \geq 1/k$.
- **Output:** A path from s to t in G.

The completeness of these two problems is given by the following theorem.

Theorem 3.1 POLY-MIXING S-T CONNECTIVITY *is complete for* $\mathbf{prRL}$. POLY-MIXING FIND PATH *is complete for* $\mathbf{searchRL}$.

Proof: First, we show that these problems are in $\mathbf{prRL}$ and $\mathbf{searchRL}$, respectively, by giving randomized logspace algorithms for them. Given an instance $(G, s, t, 1^k)$, we take a random walk of length $m = k \cdot \ln(2k)$ from s, where N is the number of vertices in G. The $\mathbf{searchRL}$ algorithm simply outputs this walk, and the $\mathbf{prRL}$ algorithm accepts if this walk ends at t. To analyze this algorithm, let $\alpha \in \mathbb{R}^N$ be the initial probability distribution, concentrated at s. Then, the probability that the walk ends at t is

$$\begin{aligned}
(M_G^m \alpha)(t) &\geq \pi_s(t) - \|M_G^m \alpha - \pi_s\|_{\pi_s} \\
&\geq \frac{1}{k} - \lambda_s(G)^m \cdot \|\alpha - \pi_s\| \\
&\geq \frac{1}{k} - \left(1 - \frac{1}{k}\right)^m \cdot 1 \\
&\geq \frac{1}{k} - \frac{1}{2k} \geq \frac{1}{2k}
\end{aligned}$$

The success probability of both algorithms can be boosted by repetitions as usual (outputting the sequence of walks taken in the case of $\mathbf{searchRL}$).

Now we show that every problem in $\mathbf{prRL}$ and $\mathbf{searchRL}$, respectively, reduce to POLY-MIXING S-T CONNECTIVITY and POLY-MIXING FIND PATH. Let M be a randomized logspace machine, running in time at most $p(n) \leq \text{poly}(n)$. Given an input x of length n for M, we construct a graph G whose vertices are of the form (i, τ), where $i \in \{1, \ldots, p(n)\}$ is a "layer", and $\tau \in \{0,1\}^{O(\log n)}$ describes a possible configuration of M (i.e. the state, the contents of the work tape, and the position of the input head). We let $s = (1, \alpha)$ where α is the unique start configuration of M, and $t = (p(n), \beta)$ where β is the (wlog unique) accepting configuration of M. (In the case of a $\mathbf{searchRL}$ algorithm, we have M accept if any of the strings it outputs satisfy the relation R.) We create four outgoing edges from each vertex (i, γ). Two edges are always self-loops. If $i = p(n)$, then the other two edges go to s. If $i < p(n)$, then the they go to vertices of the form $(i+1, \gamma')$ and $(i+1, \gamma'')$, for γ', γ'' as follows. If γ is a configuration where M reads a new random bit, then we take γ' and γ'' to be the two configurations that M would enter depending on the two possible values of the random bit. If γ is a configuration where M does not read a new random bit, then we set $\gamma' = \gamma''$ to be the unique next configuration in M's computation on x. If γ is a halting configuration of M, then we set $\gamma' = \gamma'' = \gamma$.

Let us analyze the stationary distribution and mixing time of a random walk on G. It can be verified that the following distribution π is on vertices (i, τ) is stationary for G: choose i uniformly at random from $\{1, \ldots, p(n)\}$, run M for i steps on input x, and let τ be M's configuration. We see that if $x \in L$, then $\pi(t) > 1/2p(n)$, and if $x \notin L$, then $\pi(t) = 0$. In both cases $\pi(t) = 1/p(n)$.

Because of the self-loops and the fact that $\pi(s) > 0$, we can bound $\lambda_{s,\pi}(G)$ by computing the conductance $h_\pi(G)$ and applying Lemma 2.3. To lower-bound the conductance, we need to lower bound $\Pr[X' \notin A | X \in A] = \Pr[X \in A \wedge X' \notin A] / \pi(A)$, where X is chosen according to π, X' is a random step from X, and A is any set such that $0 < \pi(A) \leq 1/2$. To bound this, we consider a random walk $X_1, \ldots, X_r$ of length $r = 3p(n)$ started in the stationary distribution π, and separate into two cases depending on whether $s \in A$.

If $s \notin A$, then the following holds:

$$\begin{aligned} r \cdot \Pr\left[X \in A \wedge X' \notin A\right] &\geq \Pr[\exists i X_i \in A \wedge X_{i+1} \notin A] \\ &\geq \Pr[X_1 \in A, s \in \{X_2, \ldots, X_r\}] \\ &\geq \pi(A) \cdot (1 - 2^{-\Omega(p(n))}) \geq \pi(A)/2, \end{aligned}$$

where the last inequality holds because a random walk of length $3p(n)$ (from any vertex) visits s with probability $1 - 2^{-\Omega(p(n))}$, as every edge out of layer $p(n)$ goes to s and every step moves to the next layer with probability at least $1/2$.

If $s \in A$, then the following holds:

$$\begin{aligned} r \cdot \Pr\left[X \in A \wedge X' \notin A\right] &= r \cdot \Pr\left[X \notin A \wedge X' \in A\right] \\ &\geq \Pr[\exists i X_i \notin A \wedge X_{i+1} \in A] \\ &\geq \Pr[X_1 \notin A, s \in \{X_2, \ldots, X_r\}] \\ &\geq \pi(\overline{A}) \cdot (1 - 2^{-\Omega(p(n))}) \\ &\geq \pi(A) \cdot (1 - 2^{-\Omega(p(n))}) \geq \pi(A)/2 \end{aligned}$$

Thus, we conclude that $\Pr[X' \notin A | X \in A] \geq 1/(2r) = 1/(6p(n))$ for every A such that $0 < \pi(A) \leq 1/2$, and hence $h_\pi(G) \geq 1/(6p(n))$. By Lemma 2.3, $\lambda_{s,\pi} \leq 1 - 1/(2 \cdot (6p(n))^2)$.

To conclude, in our reduction, we output $(G, s, t, 1^k)$, where $k = 72p(n)^2$. From the analysis above, this gives a logspace reduction from any problem in $\mathbf{prRL}$ to POLY-MIXING S-T CONNECTIVITY. Similarly,

it gives a reduction from any problem in $\mathbf{searchRL}$ to FIND PATH, because with one-way access to any path from s to t in G, in logspace we can construct polynomially many computation paths of M, at least one of which is accepting, and this in turn, can be used to obtain a polynomially many strings $y_1, \ldots, y_\ell$ at least of which is in $R(x)$. ■

We note that the above proof can be modified to give a complete problem for $\mathbf{prBPL}$, specifically where the NO instances are replaced with instances such that $\lambda_s(G) \leq 1 - 1/k$, $\pi_s(s) \geq 1/k$ and $\pi_s(t) \leq 1/2k$.

We also note that, following [AKL$^+$], the randomized algorithm for POLY-MIXING S-T CONNECTIVITY also gives a nonconstructive existence proof of polynomial-length universal traversal sequences for the corresponding class of graphs.

Proposition 3.2 *There is a polynomial p such that for every N, D, k, there exists a sequence $\psi \in [D]^{p(N,D,k)}$ such that for every N-vertex labelled directed graph G of outdegree D and vertex s in G such that $\lambda_s(G) \leq 1 - 1/k$, following the walk ψ from s visits all vertices v of G for which $\pi_s(v) \geq 1/k$.*

4 Operations on Directed Graphs

Recall that, in this paper, we allow graphs to be directed, have multiple edges, and have self-loops. The edges leaving and entering each vertex will typically labelled. Such a labelled graph with N vertices, maximum outdegree D_{out} and maximum indegree D_{in} can be specified by a **rotation map** $\text{Rot}_G : [N] \times [D_{\text{out}}] \to ([N] \times [D_{\text{in}}]) \cup \{\perp\}$, where $\text{Rot}_G(v, i) = (u, j)$ if the i'th edge leaving v is the j'th edge entering u, and $\text{Rot}_G(v, i) = \perp$ if V has fewer than i edges leaving it.

In this section, we define several operations on directed labelled graphs that are specified by rotation maps. For readability, we often omit the $\perp$ from the notation.

The first operation simply replaces the edge set with all walks of length t in the graph.

Definition 4.1 (powering) *Let G be a labelled graph given by rotation map $\text{Rot}_G : [N] \times [D] \to [N] \times [B]$. The t'**th power** of G is the graph G^t with rotation map is given by $\text{Rot}_{G^t} : [N] \times [D]^t \to [N] \times [B]^t$ defined by $\text{Rot}_{G_t}(v_0, (k_1, k_2, \ldots, k_t)) = (v_t, (\ell_t, \ell_{t-1}, \ldots, \ell_1))$, where these values are computed via the rule $(v_i, \ell_i) = \text{Rot}_G(v_{i-1}, k_i)$ (and if any of these evaluations yield $\perp$, then the final output is also $\perp$).*

Lemma 4.2 *If G has spectral expansion λ with respect to vertex s and stationary distribution π, then G^t has spectral expansion λ^t with respect to vertex s and stationary distribution π.*

For simplicity, the next few definitions restrict to rotation maps where the outdegree bound D is the same as the indegree bound B.

In the replacement product, we combine a graph G_1 with N_1 vertices and a rotation map of degree D_1 with a graph G_2 that has D_2 vertices and a rotation map of degree D_2. The product graph has $D_1 N_1$ vertices, that we think of as being grouped into N_1 "clouds" of size D_1, one cloud for each vertex of G_1. Each cloud is a copy of the graph G_2. In addition, if the i-th outgoing edge from vertex v in G_1 was the j-th incoming edge in w (that is, if $\text{Rot}_(G_1)(v, i) = (w, j)$, then, in the product graph, there is an edge from the i-th vertex in the cloud of v to the j-th vertex in the cloud of w. The formal definition follows.

Definition 4.3 (replacement product) *If G_1 is a labelled graph on N_1 vertices with rotation map $\text{Rot}_{G_1} : [N_1] \times [D_1] \to [N_1] \times [D_1]$ and G_2 is a labelled graph on D_1 vertices with rotation map $\text{Rot}_{G_2} : [D_1] \times [D_2] \to [D_1] \times [D_2]$, then their* **replacement product** *$G_1 \textcircled{r} G_2$ is defined to be the graph on $[N_1] \times [D_1]$ vertices whose rotation map $\text{Rot}_{G_1 \textcircled{r} G_2} : ([N_1] \times [D_1]) \times [D_2 + 1] \to ([N_1] \times [D_1]) \times [D_2 + 1]$ is as follows:*

$\mathrm{Rot}_{G_1 ⓡ G_2}((v,k),i)$**:**

1. *If* $i \le D_2$*, let* $(m,j) = \mathrm{Rot}_{G_2}(k,i)$ *and output* $((v,m),j)$.
2. *If* $i = D_2 + 1$*, output* $(\mathrm{Rot}_{G_1}(v,k),i)$.
3. *If the computation of* Rot_{G_2} *or* Rot_{G_1} *yields* $\perp$*, then the output is* $\perp$.

A variant, called the **balanced replacement product** in [RVW], gives equal weight to the edges coming from G_1 and from G_2, by duplicating edges that go between clouds (ie edges of the type 2) D_2 times, for a total degree of $2D_2$.

The zig-zag product, introduced in [RVW], combines, as before, a graph G_1 with N_1 vertices and a rotation map of degree D_1 with a graph G_2 that has D_1 vertices and degree D_2. The product graph has $N_1 D_1$ vertices as in the replacement product, but now there is an edge between two vertices if there is a length-three path in the replacement product graph between them, and the middle edge in the path crosses between two clouds. In particular, the degree of the zig-zag product graph is D_2^2, instead of $D_2 + 1$. The formal definition is below.

Definition 4.4 (zig-zag product [RVW]) *If* G_1 *is a labelled graph on* N_1 *vertices with rotation map* $\mathrm{Rot}_{G_1} : [N_1] \times [D_1] \to [N_1] \times [D_1]$ *and* G_2 *is a labelled graph on* D_1 *vertices with rotation map* $\mathrm{Rot}_{G_2} : [D_1] \times [D_2] \to [D_1] \times [D_2]$*, then their* **zig-zag** *product* $G_1 ⓩ G_2$ *is defined to be the graph on* $[N_1] \times [D_1]$ *vertices whose rotation map* $\mathrm{Rot}_{G_1 ⓩ G_2} : ([N_1] \times [D_1]) \times [D_2^2] \to ([N_1] \times [D_1]) \times [D_2^2]$ *is as follows:*

$\mathrm{Rot}_{G_1 ⓩ G_2}((v,k),(i,j))$**:**

1. *Let* $(k',i') = \mathrm{Rot}_{G_2}(k,i)$.
2. *Let* $(w,\ell') = \mathrm{Rot}_{G_1}(v,k')$.
3. *Let* $(\ell,j') = \mathrm{Rot}_{G_2}(\ell',j)$.
4. *Output* $((w,\ell),(j',i'))$.

5 Biregular Graphs

BIREGULAR S-T CONNECTIVITY and BIREGULAR FIND PATH are the problems obtained by restricting S-T CONNECTIVITY and FIND PATH to biregular graphs. There is no additional promise. It is not difficult to see that BIREGULAR S-T CONNECTIVITY reduces to UNDIRECTED S-T CONNECTIVITY, simply by making all edges undirected (i.e., by bidirecting all edges). Whether or not s and t are connected is maintained because, in a biregular graph, every cut has the same number of edges crossing in both directions. However, note that this is *not* a reduction from BIREGULAR FIND PATH to UNDIRECTED FIND PATH. Nevertheless, here we give a logspace algorithm for BIREGULAR FIND PATH by generalizing the ideas underlying Reingold's algorithm [Rei] to the directed case. We also obtain universal traversal sequences for bi-regular with "consistent" labelling and generators of pseudorandom walks on such graphs. We discuss both below.

Theorem 5.1 BIREGULAR FIND PATH *is in* $\mathbf{searchL}$.

5.1 Basic Facts

In a biregular graph of degree D, the rotation map $\mathrm{Rot}_G : [N] \times [D] \to [N] \times [D]$ is a permutation. Note that the uniform distribution is a stationary distribution of a random walk on a biregular graph. Thus, in this section we always refer to spectral expansion with respect to the uniform distribution. Also, in biregular graphs, if there is a path from u to v, then there is also a path from v to u, so the spectral expansion $\lambda_s(G)$ is independent of the choice of the start vertex s within a given connected component. Thus, we will often omit s from the notation, just writing $\lambda(G)$, in which case it should be understood that G is connected.

First, we note that biregular graphs have nonnegligible spectral gap, which can be proven by reduction to the undirected case [AS] via [Fil].

Lemma 5.2 *For every connected, aperiodic biregular graph G, $\lambda(G) \leq 1 - \Omega(1/DN^2)$.*

5.2 Zig-zag Product

In this section, we generalize the Zig-Zag Theorem of [RVW] to biregular graphs:

Theorem 5.3 *If $\lambda(G_1) \leq \lambda_1$ and $\lambda(G_2) \leq \lambda_2$, then $\lambda(G_1 \textcircled{z} G_2) \leq f(\lambda_1, \lambda_2)$, where*

$$f(\lambda_1, \lambda_2) \leq \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2 + \lambda_2^4},$$

and

$$f(1 - \varepsilon_1, 1 - \varepsilon_2) \leq 1 - \varepsilon_1\varepsilon_2/8.$$

We have not attempted to optimize these bounds, and they can certainly be improved. For the purposes of this paper, we only need the second bound (specifically, its consequence that for some constant ε_2, $f(1 - \varepsilon_1, 1 - \varepsilon_2) = 1 - \Omega(\varepsilon_1)$).

As noted in [RVW], the above bound on the expansion of the zig-zag product also implies a bound on the expansion of the replacement product and balanced replacement product, because the cubes of the latter graphs contain the zig-zag as a regular subgraph. Specifically, for the replacement product we obtain a bound like the above with $f(1 - \varepsilon_1, 1 - \varepsilon_2) \leq 1 - \Omega(\varepsilon_1\varepsilon_2/D_2)$, where D_2 is the degree of G_2, and for the balanced replacement we get $f(1 - \varepsilon_1, 1 - \varepsilon_2) \leq 1 - \Omega(\varepsilon_1\varepsilon_2)$.

Our algorithm, like [Rei], we will only use the following consequence of the second bound above: if G_2 is a good expander in the sense that $\lambda(G_2)$ is bounded by a constant less than 1 and $\lambda(G_1) \leq 1 - \varepsilon_1$, then $\lambda(G_1 \textcircled{z} G_2) \leq 1 - \Omega(\varepsilon_1)$. We note that our proof of this $1 - \Omega(\varepsilon_1)$ bound is significantly simpler than the previous proofs of this bound in the undirected case, for either the zig-zag or replacement products.[5]

Proof: (of Theorem 5.3) Let M be the transition matrix of the random walk on $G_1 \textcircled{z} G_2$. According to Definition 2.1, we must show that, for every vector $w \in \mathbb{R}^{N_1 \cdot D_1}$ orthogonal to the uniform distribution $u_{N_1 D_1}$, Mw is shorter than w by a factor of $f(\lambda_1, \lambda_2)$.[6]

[5] The basic analysis of the undirected zig-zag product in [RVW] only gives a bound of $1 - \Omega(\varepsilon_1^2)$. Only a much more complicated and less intuitive analysis, that uses the undirectedness of G_1 in additional ways, gives the $1 - \Omega(\varepsilon_1)$ bound. The Martin–Randall [MR2] decomposition theorem for Markov chains also implies a $1 - \Omega(\varepsilon_1)$ bound for the undirected replacement products, but its full proof (relying on [CPS]) is also fairly involved.

[6] Here we already depart from the analysis of [RVW], which bounds $\max_w \langle |\langle Mw, w\rangle| / \langle w, w\rangle$, a quantity that is equal to $\max_w \|Mw\|/\|w\|$ in case M is symmetric, but not in general. The bound $f(1 - \varepsilon_1, 1 - \varepsilon_2) = 1 - \Omega(\varepsilon_1\varepsilon_2)$ in [RVW] uses symmetry in additional ways, e.g. to deduce that the matrix $\tilde{A}$ defined below is a reflection in $\mathbb{R}^{N_1}$.

Following [RVW], we relate M to the transition matrices of G_1 and G_2, which we denote by A and B, respectively. First, we decompose M into the product of three matrices, corresponding to the three steps in the definition of $G_1 ⓩ G_2$'s edges. Let $\tilde{B}$ be the transition matrix for taking a random G_2-step on the second component of $[N_1] \times [D_1]$, i.e. $\tilde{B} = I_{N_1} \otimes B$, where I_{N_1} is the $N_1 \times N_1$ identity matrix. Let $\tilde{A}$ be the permutation matrix corresponding to Rot_{G_1}. By the definition of $G_1 ⓩ G_2$, we have $M = \tilde{B}\tilde{A}\tilde{B}$.

For any vector $v \in \mathbb{R}^{N_1 \cdot D_1}$, we define $v^{\|}$ to be the component of v that is constant on every cloud. That is, $v^{\|} = \overline{v} \otimes u_{D_1}$, where $\overline{v} \in \mathbb{R}^{N_1}$ is defined by $\overline{v}_v = \sum_{i \in [D_1]} v_{v,i}$. We define $v^{\perp} = v - v^{\|}$, so $v^{\perp}$ is orthogonal to uniform on every cloud. Since $v^{\perp} \perp v^{\|}$, we have $\|v\|^2 = \|v^{\|}\|^2 + \|v^{\perp}\|^2$.

Now, recall that our aim is to show that $\|Mw\| \leq f(\lambda_1, \lambda_2) \cdot \|w\|$ for every $w \perp u_{N_1 D_1}$. In light of the decomposition $M = \tilde{B}\tilde{A}\tilde{B}$, we define $x = \tilde{B}w$, $y = \tilde{A}x$, $z = \tilde{B}y$. Now, we make several observations about the effect of these transformations:

- Since u_{D_1} is the stationary distribution for $\tilde{B}$, we have $x^{\|} = w^{\|}$, $x^{\perp} = \tilde{B}w^{\perp}$, $z^{\|} = y^{\|}$, and $z^{\perp} = \tilde{B}y^{\perp}$.
- By the expansion of G_2, we have $\|x^{\perp}\| \leq \lambda_2 \cdot \|w^{\perp}\|$ and $\|z^{\perp}\| \leq \lambda_2 \cdot \|y^{\perp}\|$.
- By linearity, $y^{\|} = (\tilde{A}x^{\|})^{\|} + (\tilde{A}x^{\perp})^{\|}$ and $y^{\perp} = (\tilde{A}x^{\|})^{\perp} + (\tilde{A}x^{\perp})^{\perp}$.
- By the expansion of G_1, we have $|(\tilde{A}x^{\|})^{\|}\| \leq \lambda_1 \cdot \|x^{\|}\|$.
- Since $\tilde{A}$ is a permutation matrix, we have $\|y\| = \|x\|$, and

$$\begin{aligned} 0 &= \langle x^{\|}, x^{\perp} \rangle \\ &= \langle \tilde{A}x^{\|}, \tilde{A}x^{\perp} \rangle \\ &= \langle (\tilde{A}x^{\|})^{\|} + (\tilde{A}x^{\|})^{\perp}, (\tilde{A}x^{\perp})^{\|} + (\tilde{A}x^{\perp})^{\perp} \rangle \\ &= \langle (\tilde{A}x^{\|})^{\|}, (\tilde{A}x^{\perp})^{\|} \rangle + \langle (\tilde{A}x^{\|})^{\perp}, (\tilde{A}x^{\perp})^{\perp} \rangle. \end{aligned}$$

  That is, $\langle (\tilde{A}x^{\|})^{\|}, (\tilde{A}x^{\perp})^{\|} \rangle = -\langle (\tilde{A}x^{\|})^{\perp}, (\tilde{A}x^{\perp})^{\perp} \rangle$.

Given these observations, it is just a calculation to obtain the two desired bounds on the length of $z = Mw$.

$$\begin{aligned} \|z\|^2 &= \|z^{\|}\|^2 + \|z^{\perp}\|^2 \\ &\leq \|y^{\|}\|^2 + \lambda_2^2 \cdot \|y^{\perp}\|^2 \\ &= \|(\tilde{A}x^{\|})^{\|} + (\tilde{A}x^{\perp})^{\|}\|^2 + \lambda_2^2 \cdot \|y^{\perp}\|^2 \\ &\leq \|(\tilde{A}x^{\|})^{\|}\|^2 + \|(\tilde{A}x^{\perp})^{\|}\|^2 + 2\|(\tilde{A}x^{\|})^{\|}\| \cdot \|(\tilde{A}x^{\perp})^{\|}\| + \lambda_2^2 \cdot \|y^{\perp}\|^2 \\ &\leq \lambda_1^2 \cdot \|x^{\|}\|^2 + \|x^{\perp}\|^2 + 2\lambda_1 \|x^{\|}\| \cdot \|x^{\perp}\| + \lambda_2^2 \|x\|^2 \\ &\leq \lambda_1^2 \cdot \|w^{\|}\|^2 + \lambda_2^2 \cdot \|w^{\perp}\|^2 + 2\lambda_1 \|w^{\|}\| \cdot \lambda_2 \|w^{\perp}\| + \lambda_2^2 \cdot (\|w^{\|}\|^2 + \lambda_2^2 \cdot \|w^{\perp}\|^2) \\ &\leq (\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2 + \lambda_2^4) \cdot \|w\|^2, \end{aligned}$$

where in the last inequality we use the fact that $\|w^{\|}\|^2 + \|w^{\perp}\|^2 = \|w\|^2$.

For the second bound, we consider two cases.

Case I: $\|w^{\perp}\|^2 \geq \min\{\sqrt{1-\lambda_1^2}/(4\lambda_2), 1/2\} \cdot \|w\|^2$.

$$\begin{aligned}
1-\|z\|^2 &\geq 1-\|x\|^2 \\
&\geq 1-\left(\|w^{\|}\|^2+\lambda_2^2\cdot\|w^{\perp}\|^2\right) \\
&= (1-\lambda_2^2)\cdot\|w^{\perp}\|^2 \\
&\geq \min\{(1-\lambda_2^2)\cdot\sqrt{1-\lambda_1^2}/(4\lambda_2), (1-\lambda_2^2)/2\}\cdot\|w\|^2
\end{aligned}$$

Case II: $\|w^{\perp}\| < \min\{\sqrt{1-\lambda_1^2}/8\lambda_2, 1/\sqrt{2}\}\cdot\|w\|$.

For notational convenience, we set δ_1 so that $\delta_1^2 = \|(\tilde{A}x^{\|})^{\perp}\|^2/\|x^{\|}\|^2 \geq 1-\lambda_1^2$.

$$\begin{aligned}
1-\|z\|^2 &\geq (1-\lambda_2^2)\cdot\|y^{\perp}\|^2 \\
&= (1-\lambda_2^2)\cdot\|(\tilde{A}x^{\|})^{\perp}+(\tilde{A}x^{\perp})^{\perp}\|^2 \\
&= (1-\lambda_2^2)\cdot\left[\|(\tilde{A}x^{\|})^{\perp}\|^2+\|(\tilde{A}x^{\perp})^{\perp}\|^2+2\langle(\tilde{A}x^{\|})^{\perp},(\tilde{A}x^{\perp})^{\perp}\rangle\right] \\
&\geq (1-\lambda_2^2)\cdot\left[\delta_1^2\cdot\|x^{\|}\|^2+0-2\delta_1\cdot\|x^{\|}\|\cdot\|x^{\perp}\|\right] \\
&\geq (1-\lambda_2^2)\cdot\left[\delta_1^2\cdot\|w^{\|}\|^2-2\delta_1\cdot\|w^{\|}\|\cdot\lambda_2\|w^{\perp}\|\right] \\
&\geq (1-\lambda_2^2)\cdot\left[3\delta_1^2/4-2\delta_1\lambda_2\cdot\sqrt{1-\lambda_1^2}/(4\lambda_2)\right]\cdot\|w\|^2 \\
&\geq [(1-\lambda_2^2)\cdot(1-\lambda_1^2)/4]\cdot\|w\|^2.
\end{aligned}$$

Thus, we conclude that

$$\begin{aligned}
1-f(\lambda_1,\lambda_2) &\geq (1-f(\lambda_1,\lambda_2)^2)/2 \\
&\geq \frac{1}{2}\cdot\min\left\{\frac{(1-\lambda_2^2)\cdot\sqrt{1-\lambda_1^2}}{4\lambda_2}, \frac{1-\lambda_2^2}{2}, \frac{(1-\lambda_2)^2\cdot(1-\lambda_1^2)}{4}\right\} \\
&\geq \frac{1}{2}\min\left\{\frac{\varepsilon_2\cdot\sqrt{\varepsilon_1}}{4(1-\varepsilon_2)}, \frac{\varepsilon_2}{2}, \frac{\varepsilon_2\cdot\varepsilon_1}{4}\right\} \\
&= \varepsilon_1\varepsilon_2/8,
\end{aligned}$$

as desired. ■

5.3 The Path Finding Algorithm

We have seen that powering and the zig-zag graph product has essentially the same affect on biregular graphs as on undirected graphs. Therefore, both the decision and search versions of the st-connectivity algorithm of [Rei] can be extended (without any substantial change) to biregular graphs. This implies Theorem 5.1, which states that BIREGULAR FIND PATH is in $\mathbf{searchL}$. As the algorithm here is essentially the same as in [Rei], we only provide a sketch of the proof.

Proof Sketch: [of Theorem 5.1] We describe a log-space algorithm $\mathcal{A}$ that gets as input a D-biregular (i.e. both the indegree and the outdegree of each vertex is D) graph G on N-vertices and two vertices s and t and outputs a path from s to t if such a path exists (otherwise, it will output 'not connected').

The algorithm will rely on a constant size (undirected) expander graph H, given by its rotation map Rot_H, with rather weak parameters. More specifically, H will be D_{e}-regular, for some constant D_{e}, it will have $(D_{\mathsf{e}})^{80}$ vertices (no attempt was made to optimize the constants), and $\lambda(H) \leq 1/2$. The expander H can be obtained via exhaustive search or any one of various explicit constructions.

The first step of the algorithm, will be to reduce the input G, s, t into a new input G_{reg}, s', t' where G_{reg} is $(D_{\mathsf{e}})^{80}$-biregular on $N \cdot D$ vertices, every connected component of G_{reg} is aperiodic, and s and t are connected in G if and only if s' and t' are connected in G_{reg}. Furthermore, a path from s' to t' in G_{reg} can be translated in log-space into a path from s to t in G. The reduction itself is quite standard: Each vertex of G is replaced with a cycle with D vertices. In addition, the i^{th} vertex (v, i) in the cycle that corresponds to v is connected to $(w, j) = \mathrm{Rot}_G(v, i)$ in the cycle that corresponds to w. Up to now, both the indegree and the outdegree of each vertex is three. Therefore, we add to each vertex $(D_{\mathsf{e}})^{80} - 3$ self loops (this also guarantees that each connected component of G_{reg} is aperiodic). The vertices s' and t' are arbitrary vertices from the cycles that correspond to s and t. A path from s' to t' in G_{reg} can easily be projected down to a path from s to t in G.

The next step is a reduction of G_{reg}, s', t' to a new input $G_{\mathsf{exp}}, s'', t''$ of BIREGULAR FIND PATH, such that each connected component of G_{exp} is an expander (and in particular has a logarithmic diameter), and s' and t' are connected in G_{reg} if and only if s'' and t'' are connected in G_{exp}. Furthermore, this is a log-space reduction and a path from s'' to t'' in G_{exp} can be translated in log-space into a path from s' to t' in G_{reg}. This step is the heart of the algorithm, and it essentially completes the algorithm. All that is left to do is enumerate all logarithmically-long paths from s'' in G_{exp} and output one of them if it reaches t'' (after translating it in two steps to a path from s to t in G).

The transformation from G_{reg} to G_{exp} is defined recursively. Set G_0 to equal G_{reg}, and for $i > 0$ define G_i recursively by the rule:

$$G_i = (G_{i-1} \textcircled{z} H)^{40}.$$

Finally, define $G_{\mathsf{exp}} = G_\ell$ for $\ell = O(\log(N \cdot D))$ (that will be determined by the analysis). It follows inductively that each G_i is a $(D_{\mathsf{e}})^{80}$-biregular graph over $[N] \times [D] \times [(D_{\mathsf{e}})^{80}]^i$. In particular, the zig-zag product of G_i and H is well defined. In addition, since D_{e} is a constant, and ℓ is logarithmic then G_ℓ has $poly(N \cdot D)$ vertices.

Assume that G_{reg} is connected, then by Lemma 5.2, $\lambda(G_{\mathrm{reg}}) \leq 1 - 1/\mathrm{poly}(N \cdot D)$. By Lemma 4.2 and Theorem 5.3 (properties of powering and the zig-zag product for biregular graphs), we have that unless $\lambda(G_i)$ is already smaller than some fixed constant then $\lambda(G_i) \leq (\lambda(G_{i-1}))^2$. This means that for some $\ell = O(\log(N \cdot D))$, we have that $\lambda(G_\ell)$ is guaranteed to be smaller than some fixed constant. In other words, G_{exp} is an expander. What if G_{reg} has several connected components? Since both powering and the zig-zag product operate separately on each connected component, we have that for every $S \subseteq [N] \times [D]$, if S contains the vertices of a connected component of G_{reg} then $S \times [(D_{\mathsf{e}})^{80}]^\ell$ contains the vertices of a connected component of G_{exp}, and the subgraph of G_{exp} induced by these vertices is an expander. By this argument, it is natural to select s'' to be any vertex in $\{s'\} \times [(D_{\mathsf{e}})^{80}]^\ell$ and similarly regarding t''. This choice indeed satisfies the requirements of the reduction.

It remains to argue that the transformation of G_{reg} to G_{exp} is log-space and that a path on G_{exp} translates in log-space into a path on G_{reg}. The intuition is that taking a step on G_i translates to a constant number of operations, some of which are taking a step on G_{i-1} and the rest require constant space. As the space used for each one of these operations can be reused for the subsequent operations, the space needed to walk

on G_i is only larger by a constant than the space needed to walk on G_{i-1}. Furthermore, this evaluation in particular translates a step on G_i to a path of constant length between the corresponding vertices of G_{i-1}. The space-efficiency requirements follow by induction. □

5.4 Universal Traversal Sequences

We now show how the BIREGULAR FIND PATH algorithm described above also implies a log-space constructible universal traversal sequence for "consistently labelled" biregular graphs. Here we refer to biregular graphs of degree D in which only the outgoing edges from each vertex are numbered from 1 to D. We call such a labelling **consistent** if all the edges coming into any vertex of the graph have distinct labels, i.e. no vertex v can be both u's i^{th}-neighbor and w's i^{th}-neighbor (for any distinct vertices u and w). In other words, if we use the same labels to number the edges incoming at each vertex (if (u, v) is the i'th edge leaving u, we consider it to be the i'th edge entering v), we obtain a legal labelling of incoming edges (in that each label in $[D]$ will get used exactly once at each vertex). Every biregular graph has a consistent labelling; this is equivalent to the fact that every D-regular bipartite graph is the union of D perfect matchings. . (However, finding a consistent labelling may not be feasible in log-space.)

Consistent labelling is the weakest restriction for which efficiently constructible universal traversal sequences are known *even for undirected expander graphs* [HW]. For general graphs, in the undirected case, the st-connectivity algorithm of [Rei] directly gives efficiently constructible universal traversal sequences under stronger restriction on the labelling. So in fact, the generalization to biregular graphs is useful even from the point of view of undirected graphs. Our first step is to argue that the universal traversal sequences for expanders given by Hoory and Wigderson [HW] can be extended to the biregular case.

Definition 5.4 *Let D and N be two integers and let $\mathcal{G}$ be a subset of the labelled D-biregular connected graphs on N vertices. We say that a sequence of values in $[D]$ is a universal traversal sequence for $\mathcal{G}$, if for every graph $G \in \mathcal{G}$, and every vertex $s \in [N]$, the walk that starts in s and follows the edges of G according to the sequence of labels visits all the vertices of the graph.*

Lemma 5.5 *For every two constants D and λ where D is a positive integer and $\lambda < 1$, there exists a log-space algorithm that on input 1^N produces a universal traversal sequence for all connected, consistently labelled D-regular graphs G on N-vertices with $\lambda(G) \leq \lambda$.*

Proof Sketch: The universal traversal sequence of Hoory and Wigderson [HW] works just as well in the biregular case. The only properties used in their analysis are that (1) A walk that starts at two distinct vertices and follows the same set of labels ends in two distinct vertices (this is where the consistent labelling is used). (2) For two sets of vertices A and B one of size K and the other of size $N - K$, either the intersection $A \cap B$ or the number of edges from A to B are $\Omega(\min\{K, N - K\})$ (this is where the expansion is used). Both of these properties also hold in the biregular case. □

Theorem 5.6 *There exists a log-space algorithm that on input $1^N, 1^D$ produces a universal traversal sequence for all connected, consistently labelled D-regular graphs G on N-vertices.*

Proof Sketch: Consider some connected, consistently labelled D-regular graphs G on N-vertices. We will show a log-space algorithm $\mathcal{A}$ that produces a universal traversal sequence for $\{G\}$. We will then argue that the algorithm does not need access to G which will imply the theorem (as the output of $\mathcal{A}$ will be good for any such graph G).

The crucial observation is that, as noted above, given a consistently labelled graph G, we can assume without loss of generality that every edge (u, v) has the same label as an out going edge from u and as an incoming edge to v. Observe that, for the purpose of universal traversal sequence, the only labels that matter are the outgoing labels from each vertex (the incoming labels, which define the rotation map of the graph, are ignored during the walk - therefore any legal labelling will do). In other words, we can assume without loss of generality that whenever $(u, j) = \mathrm{Rot}_G(v, i)$ we have that $i = j$. From now on, our proof follows the same lines as the construction of universal traversal sequence in [Rei], and is therefore only sketched here.

Consider the two graphs G_{reg} and G_{exp} that are obtained from G (and *do depend* on Rot_G) in the proof of Theorem 5.1. By the analysis in that proof, G_{exp} is an expander. Furthermore, as powering and the zig-zag product preserve the property of consistent labeling, we have that G_{exp} is also consistently labelled. Lemma 5.5 now implies that there exists a universal traversal sequence $\vec{a}$ for $\{G_{\mathrm{exp}}\}$ and its log-space construction is independent of G. Now consider the walk on G_{exp}, following $\vec{a}$ from some vertex $(s, 1^{\ell+1})$, where $s \in [N]$. This walk covers all of the vertices of G_{exp}. By the construction of G_{exp}, the sequence of labels $\vec{a}$ can be translated in log-space (again, without access to G) into a sequence $\vec{b}$ of labels, such that the walk from $(s, 1)$ (for any $s \in [N]$) which follows these labels, visits all the vertices of G_{reg}.

The next step is to translate $\vec{b}$ into a universal traversal sequence for $\{G\}$. Consider the walk from $(s, 1)$ on G_{reg}. We want to simulate this walk without knowing s and without access to G. On the other hand, at each step all we want to know is a value $c \in [D]$ such that we are now at some vertex (v, c). To begin with c is set to one. It is easy update c (one up or one down) when taking a step on one of the cycles in the definition of G_{reg}. Labels that correspond to self loops can be ignored. We are left with edges that cross between two different cycles (that correspond to two vertices of G). By our assumption above, in such a case c remains unchanged. Furthermore, the values of c when an edge between cycles is taken, are exactly the labels of edges in G that are traversed by the projection on G of the walk defined by $\vec{b}$. To conclude, the sequence $\vec{c}$ is simply the sequence of values of c in the simulation described above, when edges between cycles are traversed. □

5.5 A Pseudorandom Generator

In this section we show that the path finding algorithm also implies a generator with logarithmic seed length that produces in log-space a "pseudorandom walk" for consistently labelled biregular graphs. This means that from any start vertex, following the pseudorandom walk leads to an almost uniformly distributed vertex. In other words, just as the random walk, the pseudorandom walk converges to the stationary distribution. This seems to be a result of independent interest. In particular, we show in Section 6 that a similar pseudorandom generator (or even weaker), that works for biregular graphs with *arbitrary labels*, would prove that $\mathbf{RL} = \mathbf{L}$.

The intuition for the generator is as follows. In the path-finding algorithm, an expander graph G_{exp} is constructed. In this graph a short random walk converges to the uniform distribution. As in the proof for the universal traversal sequences, the sequence of labels of the (random) walk on G_{exp} can be translated to a (pseudorandom) sequence of labels for a walk on G. Furthermore, this sequence of labels is independent of G (and can be computed in log-space without access to G). Note that all nodes of the original graph G are expanded to "clouds" of equal size. Therefore, the pseudorandom walk converges to the uniform distribution on the vertices of G (which is the projection on G of the uniform distribution on the vertices of G_{exp}). Formalizing the above arguments will indeed imply a generator that produces a pseudorandom walk of length polynomial in the size of the graph. However, a truly random walk will converge faster if G has a

larger eigenvalue gap. Theorem 5.7 takes this into account and implies, in this case, a pseudorandom walk that is shorter as well.

Theorem 5.7 *For every $N, D \in \mathbb{N}$, $\delta, \gamma > 0$, there is a generator $\mathrm{PRG} = \mathrm{PRG}_{N,D,\delta,\gamma} : \{0,1\}^r \to [D]^\ell$ with seed length $r = O(\log(ND/\delta\gamma))$, and walk length $\ell = \mathrm{poly}(1/\gamma) \cdot \log(ND/\delta)$, computable in space $O(\log(ND/\delta\gamma))$ such that for every* consistently labelled *$(N, D, 1-\gamma)$ biregular graph G and every vertex s in G, talking walk $\mathrm{PRG}(U_r)$ from s ends at a vertex that is distributed δ-close to uniform (in variation distance).*

Proof Sketch: Let G be a *consistently labelled* $(N, D, 1-\gamma)$ biregular graph and s any vertex of G. We will construct a distribution on a sequence of labels such that taking a walk from s on G according to these labels, ends at a vertex that is distributed δ-close to uniform (in variation distance). Since the distribution of labels will be independent of G and s (and will only depend on N, D, δ, and γ) this will imply a pseudorandom generator.

As in the proof of Theorem 5.1, we consider in our analysis two additional graphs G_{reg} and G_{exp}. Their definition will be slightly modified here. First, G_{reg} will be obtained by a zig-zag product (or a replacement product) with a constant degree expander on D vertices. Adding self loops we get an $(ND, (D\mathrm{e})^{80}, 1 - \Omega(\gamma))$ biregular graph. The advantage of doing that (instead of a replacement product with a cycle as in the proof of Theorem 5.1), is that the eigenvalue gap of G_{reg} is only smaller by a constant than the eigenvalue gap of G. We now define G_{exp} similarly to the proof of Theorem 5.1, by recursively applying the zig-zag product and powering. However, since we start with a stronger guarantee on the eigenvalue gap of G_{reg} we only need $\ell = O(\log(1/\gamma))$ levels of recursion to bring spectral gap to a constant. The size of the final expander G_{exp} is thus $N_{\mathrm{fin}} = N \cdot D \cdot 2^{O(\ell)} = ND \cdot \mathrm{poly}(1/\gamma)$.

Consider now a random walk of length $m_{\mathrm{fin}} = O(\log(N_{\mathrm{fin}}/\delta)) = O(\log(ND/\delta\gamma))$ in G_{exp}. Such a walk starting from any vertex in the vertices in G_{exp} which correspond to s will converge to the uniform distribution on the vertices of G_{exp}, up to variation distance δ. As in the proof of Theorem 5.6, this walk projects to a walk on G. Since the uniform distribution on vertices of G_{exp} projects to the the uniform distribution on vertices of G, we get that the walk in G also converges to the uniform distribution on the vertices of G_{exp}, up to variation distance δ. As in the proof of Theorem 5.6, we note that we can assume without loss of generality that in the rotation map of G the label of an edge (u, v) is identical both as an outgoing edge from u and as an incoming edge to v. This implies (as in that proof), that the edge labels taken by the walk on G are actually independent of G and s and can be computed in the required small space, just knowing N, D, γ, and δ.

We make the following observations:

- The randomness required is $r = O(m_{\mathrm{fin}}) = O(\log(ND/\delta\gamma))$.
- The walk length is $\ell = m_{\mathrm{fin}} \cdot 2^{O(\ell)} = \log(ND/\delta\gamma) \cdot \mathrm{poly}(1/\gamma) = \log(ND/\delta) \cdot \mathrm{poly}(1/\gamma)$.

□

6 Reducing the General Case to Biregular Graphs

In this section, we prove that if there exists a pseudorandom generators for walks on biregular graphs *whose edges are arbitrarily labelled*, then $\mathbf{RL} = \mathbf{L}$ and also $\mathbf{searchRL} = \mathbf{searchL}$. Theorem 5.7 implies a generator for walks on biregular graphs with the additional restriction that the labelling of the edges is consistent. Lifting this restriction would imply that $\mathbf{RL} = \mathbf{L}$. In fact, such a generator would also imply $\mathbf{BPL} = \mathbf{L}$. However, we concentrate in this preliminary version on the case of $\mathbf{RL}$.

Theorem 6.1 *There is a universal constant $\alpha > 0$ such that the following holds for every constant $a \in \mathbb{N}$. Suppose that for every $N, D \in \mathbb{N}$, $\delta, \gamma > 0$, there is a generator $\mathrm{PRG} = \mathrm{PRG}_{N,D,\delta,\gamma} : \{0,1\}^r \to [D]^\ell$ with seed length $r = a\log(ND/\delta\gamma)$, and walk length $\ell = (1/(\gamma\delta))^a \cdot (ND)^\alpha$, computable in space $a\log(ND/\delta\gamma)$ such that for every $(N, D, 1-\gamma)$ biregular graph $G = (V, E)$ and every vertex $s \in V$ and every subset $T \subseteq V$ of density at least δ, the walk from s following the labels $\mathrm{PRG}(U_r)$ visits T with probability at least $(\delta\gamma)^a/(ND)^\alpha$. Then $\mathbf{RL} = \mathbf{L}$ and $\mathbf{searchRL} = \mathbf{searchL}$.*

Note that the above theorem requires that the length ℓ of the pseudorandom walks have limited dependence on N and D, being bounded by $(ND)^\alpha$ rather than being polynomial or even linear in ND. Still, this is a much milder requirement than what is achieved by our generator for consistently labelled graphs (Thm. 5.7), which achieves logarithmic dependence. We also note that a pseudorandom generator for logspace algorithms with logarithmic seed length would imply the above theorem, because a truly random walk of length $O(1/\gamma) \cdot O(\log(ND/\delta))$ would end at T with probability at least $\delta/2$, and such a walk can be implemented in space $O(\log(ND/\delta\gamma))$.

Roughly speaking, we will prove Theorem 6.1 by showing that for every poly-mixing graph G, there exists a biregular graph G_{reg} such that the correctness of the generator on G_{reg} implies the correctness of (a modification of) the generator on G. The construction of G_{reg} from G is given by the following lemma. We stress that this construction is only done in the *analysis*, and thus need not be computable in log-space.

Lemma 6.2 *There is a universal constant c such that the following holds. Let $G = (V, E)$ be any d-outregular graph on n vertices with vertices $s, t \in V$ and stationary distribution π such that $\pi(s) \geq 1/k$, $\pi(t) \geq 1/k$, and $\lambda_\pi(G) \leq 1 - 1/k$. Then for every $\varepsilon > 0$, if we set $N_{\mathrm{reg}} = (ndk/\varepsilon)^c$, $D_{\mathrm{reg}} = c \cdot N_{\mathrm{reg}}/\varepsilon$, $\gamma = 1/(ndk)^c$, there is a $(N_{\mathrm{reg}}, d \cdot D_{\mathrm{reg}}, 1-\gamma)$-biregular graph G_{reg} such that the following holds. The vertex set of G_{reg} can be decomposed into "clouds" $V_{\mathrm{reg}} = \bigcup_{v \in V} C_v$ with $|C_s|, |C_t| \geq |V_{\mathrm{reg}}|/2k$. There is a bad set of edge labels $B \subseteq [d] \times [D_{\mathrm{reg}}]$ of density ε such that for every $u \in V$, vertex $\hat{u} \in C_u$ and edge label $(i, j) \in ([d] \times [D_{\mathrm{reg}}]) \setminus B$, the (i, j)'th neighbor of $\hat{u}$ in G_{reg} is in cloud C_v where v is the i'th neighbor of u in G_{reg}.*

Before proving this lemma, let's see how it implies Theorem 6.1.

Proof of Theorem 6.1: Let $(G, s, t, 1^k)$ be any instance of POLY-MIXING FIND PATH, where G is d-outregular, has n vertices, and has (promised) stationary distribution π with $\pi(s), \pi(t) \geq 1/k$ and $\lambda_\pi(G) \leq 1 - 1/k$. Set $\delta = 1/2k$, and $\varepsilon = 1/(ndk)^b$ for a large constant b to be specified later, and let $N_{\mathrm{reg}} = (ndk/\varepsilon)^c$, $D_{\mathrm{reg}} = c \cdot N_{\mathrm{reg}}/\varepsilon$, $\gamma = 1/(ndk)^c$ be the parameters of the biregular graph guaranteed by Lemma 6.2. Let $\mathrm{PRG} = \mathrm{PRG}_{N_{\mathrm{reg}}, d \cdot D_{\mathrm{reg}}, \delta, \gamma} : \{0,1\}^r \to ([d] \times [D_{\mathrm{reg}}])^\ell$ be the generator hypothesized in Theorem 6.1, with seed length $r = a\log(N_{\mathrm{reg}} D_{\mathrm{reg}}/\delta\gamma) = O(abc\log(ndk))$. and walk length

$$\ell = (1/\gamma\delta)^a \cdot (N_{\mathrm{reg}} \cdot dD_{\mathrm{reg}})^\alpha = (ndk)^{O(ac)} \cdot (ndk/\varepsilon)^{O(\alpha c)} = (ndk)^{O(ac)}/\varepsilon^{O(\alpha c)}$$

Without loss of generality we may assume that each component in $\mathrm{PRG}(U_r)$ is uniformly distributed in $[d] \times [D_{\mathrm{reg}}]$. (Shift each component of the output by adding a random element $s \leftarrow [d] \times [D_{\mathrm{reg}}]$. This only increases the seed length by a constant factor and preserves the pseudorandomness of the output because it is equivalent to shifting all labels in the biregular graph by $-s$.)

The algorithm for POLY-MIXING FIND PATH works as follows. We enumerate the $2^r = (nkd)^{O(abc)}$ seeds of PRG, for each obtaining a walk $\hat{w} \in ([d] \times [D_{\mathrm{reg}}])^\ell$ of length $\ell = (nkd)^{O(abc)}$. Taking the first components of each step in $\hat{w}$, we obtain an induced walk $w \in [d]^\ell$, which we perform on G, starting at s. If any of these walks end at t, we output that walk.

To analyze this algorithm, we consider a walk $\hat{w} \leftarrow \mathrm{PRG}(U_r)$ taken in G_{reg}, starting at any vertex of C_s. Since $\lambda(G_{\mathrm{reg}}) \leq 1-\gamma$, C_t has density at least $1/2k$, and $\delta = 1/2k$, such a walk will end in C_t with probability at least

$$(1/\delta\gamma)^a \cdot (N_{\mathrm{reg}} \cdot dD_{\mathrm{reg}})^\alpha = \varepsilon^{O(\alpha c)}/(ndk)^{O(ac)}.$$

We now argue that the induced walk w in G will end at t with nearly the same probability. By the properties of G_{reg}, this will be the case provided the walk $\hat{w}$ does not use any edge label from B. Since B has density at most ε and each edge label in $\hat{w}$ is uniformly distributed, the probability any label from B is used is at most

$$\ell \cdot \varepsilon = (ndk)^{O(ac)} \cdot \varepsilon^{1-O(\alpha c)}.$$

Thus the walk w in G ends at t with probability at least

$$\frac{\varepsilon^{O(\alpha c)}}{(ndk)^{O(ac)}} - (ndk)^{O(ac)} \cdot \varepsilon^{1-O(\alpha c)} > 0,$$

provided $\alpha \leq c/\kappa$ and $\varepsilon \leq (1/ndk)^b$ for a $b > \kappa ac$, where κ is a sufficiently large universal constant. In particular, there exists a seed of PRG that will produce a walk from s to t. ■

On mixing vs. hitting

For the proof Lemma 6.2 we will need the following two lemmas, showing that polynomial mixing time (i.e. inverse polynomial spectral gap) is equivalent to the existence of a vertex s such that a random walk of polynomial length from any vertex hits s with high probability.

Lemma 6.3 *Let G be a directed graph in which at least half of the edges leaving every vertex are self-loops. Suppose there is a vertex s and a number $\ell \in \mathbb{N}$ such that from every vertex v reachable from s, a random walk of length ℓ from v visits s with probability at least $1/2$. Then G has a stationary distribution π such that $\lambda_{s,\pi}(G) \leq 1 - 1/8\ell^2$ and $\pi(s) \geq 1/2\ell$. Moreover, a random walk from s of length $O(\ell^2 \log(\ell/\varepsilon))$ ends at a vertex distributed ε-close to π (in variation distance).*

Proof: Let G' be the induced subgraph of G consisting of all vertices reachable from s. Let π be a stationary distribution of the random walk on on G'. We bound $\lambda_{s,\pi}(G)$ (and $\pi(s)$) via conductance in a manner similar to the proof of Theorem 3.1. Because of the self-loops and the fact that $\pi(s) > 0$ (since every vertex in G' has a path to s), we can bound $\lambda_{s,\pi}(G)$ by computing the conductance $h_\pi(G)$ and applying Lemma 2.3. To lower-bound the conductance, we need to lower bound $\Pr[X' \notin A | X \in A] = \Pr[X \in A \wedge X' \notin A]/\pi(A)$, where X is chosen according to π, X' is a random step from X, and A is any set such that $0 < \pi(A) \leq 1/2$. To bound this, we consider a random walk $X_1, \ldots, X_\ell$ of length ℓ started in the stationary distribution π, and separate into two cases depending on whether $s \in A$.

If $s \notin A$, then the following holds:

$$\begin{aligned} \ell \cdot \Pr\left[X \in A \wedge X' \notin A\right] &\geq \Pr[\exists i X_i \in A \wedge X_{i+1} \notin A] \\ &\geq \Pr[X_1 \in A, s \in \{X_2, \ldots, X_\ell\}] \\ &\geq \pi(A) \cdot (1/2), \end{aligned}$$

where the last inequality holds because a random walk of length ℓ (from any vertex in G') visits s with probability at least $1/2$ by hypothesis.

If $s \in A$, then the following holds:

$$\begin{aligned}
\ell \cdot \Pr\left[X \in A \wedge X' \notin A\right] &= \ell \cdot \Pr\left[X \notin A \wedge X' \in A\right] \\
&\geq \Pr\left[\exists i X_i \notin A \wedge X_{i+1} \in A\right] \\
&\geq \Pr\left[X_1 \notin A, s \in \{X_2, \ldots, X_\ell\}\right] \\
&\geq \pi(\overline{A})/2 \\
&\geq \pi(A)/2
\end{aligned}$$

Thus, we conclude that $\Pr\left[X' \notin A | X \in A\right] \geq 1/(2\ell)$ for every A such that $0 < \pi(A) \leq 1/2$, and hence $h_\pi(G) \geq 1/(2\ell)$. By Lemma 2.3, $\lambda_{s,\pi} \leq 1 - 1/(2 \cdot (2\ell)^2)$.

To lower bound $\pi(s)$, we note that the expected number of times s is visited in $X_1, \ldots, X_\ell$ equals $\pi(s) \cdot \ell$ on one hand, and is at least $1/2$ on the other. Thus $\pi(s) \cdot \ell \geq 1/2$.

Let π_0 be the distribution concentrated at s, $M = M_G$, and $\lambda = \lambda_{s,\pi}$. We are interested in the variation distance between $M^t\pi_0$ and π, which is smaller than their ℓ_1 distance, which equals:

$$\begin{aligned}
|M^t\pi_0 - \pi|_1 &= \sum_x \frac{|(M^t\pi_0)(x) - \pi(x)|}{\sqrt{\pi(x)}} \cdot \sqrt{\pi(x)} \\
&\leq \left(\sum_x \frac{|(M^t\pi_0)(x) - \pi(x)|^2}{\pi(x)}\right)^{1/2} \cdot \left(\sum_x \pi(x)\right)^{1/2} \\
&= \|M^t\pi_0 - \pi\|_\pi \cdot 1 \\
&\leq \lambda^t \cdot \|\pi_0 - \pi\|_\pi \\
&\leq \lambda^t \cdot \left(\frac{(1-\pi(s))^2}{\pi(s)} + \sum_{w \neq s} \frac{\pi(w)^2}{\pi(w)}\right) \\
&\leq \lambda^t \cdot (1/\pi(s)) \\
&\leq (1 - 1/(8\ell)^2)^t \cdot (2/\ell) \leq \varepsilon
\end{aligned}$$

for $t = O(\ell^2 \cdot \log(\ell/\varepsilon))$. ■

Lemma 6.4 *Let G be a directed graph on N vertices in which each vertex has outdegree at most D. Suppose that G has a stationary distribution π and a vertex s such that $\lambda_{s,\pi}(G) \leq 1 - \gamma$ and $\pi(s) \geq p$. Then for every vertex v reachable from s, a random walk of length $\ell = O((N/\gamma p) \cdot \log D)$ visits s with probability at least $1/2$.*

Note that there always exists a vertex s such that $\pi(s) \geq 1/N$, in which case ℓ is indeed polynomial.

Proof: First we bound the probability q that a random walk from v of length t ends at s. Letting $M = M_G$, $\lambda = \lambda_{s,\pi}(G) \leq 1 - \gamma$, and π_0 be the probability distribution concentrated at v, we have:

$$\begin{aligned}
|\pi(s) - q| &\leq \pi(s) \cdot \|M^t\pi_0 - \pi\|_\pi \\
&\leq \pi(s) \cdot \lambda^t \cdot \|\pi_0 - \pi\|_\pi
\end{aligned}$$

$$
\begin{aligned}
&\leq \pi(s) \cdot \lambda^t \cdot \left(\frac{(1-\pi(v))^2}{\pi(v)} + \sum_{w \neq v} \frac{\pi(w)^2}{\pi(w)} \right) \\
&\leq \pi(s) \cdot \lambda^t \cdot (1/\pi(v)) \\
&\leq \pi(s) \cdot (1-\gamma)^t \cdot D^N \\
&\leq \pi(s)/2,
\end{aligned}
$$

for $t = O((N/\gamma) \cdot \log D)$. Thus a random walk of length t ends at s with probability $q \geq \pi(s)/2 \geq p/2$. Taking $O(1/p)$ such walks in succession (for a total walk length of $\ell = O(t/p)$), we visit s with probability at least $1/2$. ■

Defining the biregular graph G_{reg}

Proof of Lemma 6.2: Let n be the number of vertices in G, d the out-degree of G, and $\pi = \pi_s$ be the stationary distribution of G (actually the induced subgraph on vertices reachable from s). By adding self-loops and applying Lemma 6.4, we may assume that G has the following properties:

1. $\pi(s) \geq 1/k$, $\pi(t) \geq 1/k$.
2. At least half of the edges leaving each vertex are self-loops.
3. For any vertex v reachable from s, a random walk of length $\ell = O(n \cdot k^2 \cdot \log d)$ from v visits s with probability at least $1/2$.

The desired biregular graph G_{reg} will essentially be a blow-up of G, with each vertex of G repeated a number of times proportional to its stationary probability, with small "corrections" to remove low-probability vertices and to fix slight irregularities (due to round-off errors).

We construct G_{reg} in several phases.

Step 1: Make all state probabilities nonnegligible. Let ε be the given error parameter. Without loss of generality, we will assume that $\varepsilon < 1/\text{poly}(n, k, d, \ell)$ for a polynomial to be specified later. Then let $D' = \text{poly}(n, \ell, 1/\varepsilon)$ for a polynomial to be specified later. Define a graph $G' = (V, E')$ on the same vertex set as G, but with degree $d \cdot D'$. For every vertex v and edge label $(i, i') \in [d] \times [D']$, we set the (i, i')'th neighbor of v in G' to be the i'th neighbor of v in G, except that we modify up to n of the edges leaving s in order to ensure that every vertex reachable from s has at least one incoming edge directly from s. (The edges to modify should be chosen so as to maintain the property that at least half of the edges from s are self-loops.) Thus a random step on G' is identical to a random step on G, except with probability at most n/D' when at vertex s.

Observe that Property 3 of G also holds in G', because any walk from a vertex v in G that visits s also visits s in G'. Thus, by Lemma 6.3, we have $\lambda_{s,\pi'}(G') \leq 1 - 1/8\ell^2$ for some stationary distribution π'. Moreover, if we take $\ell' = O(\ell^2 \log(\ell/\varepsilon))$, then a random walk of length ℓ' from s in G (resp., G') ends at a vertex distributed ε-close to π (resp., π'). Thus,

$$
\begin{aligned}
\pi'(t) &\geq \Pr[\text{r.w. in } G' \text{ of length } \ell' \text{ from } s \text{ ends at } t] - \varepsilon \\
&\geq \Pr[\text{r.w. in } G \text{ of length } \ell' \text{ from } s \text{ ends at } t] - \ell' \cdot (n/D') - \varepsilon \\
&\geq \pi(t) - \varepsilon - \varepsilon - \varepsilon \\
&\geq 1/2k,
\end{aligned}
$$

provided we take $\varepsilon \leq 1/6k$ and $D' \geq \ell' n/\varepsilon$. Similarly, we have $\pi'(s) \geq 1/2k$. And for every vertex v reachable from s, we have $\pi'(v) \geq (1/2k) \cdot (1/D')$ since there is at least one edge from s to v.

To summarize, we have established the following properties of $G' = (V, E')$:

1. For any vertex v reachable from s, a random walk of length $\ell = O(n \cdot k^2 \cdot \log d)$ from v visits s with probability at least $1/2$.
2. $\lambda_{s,\pi'}(G') \leq 1 - 1/8\ell^2$.
3. $\pi'(s) \geq 1/2k$, $\pi'(t) \geq 1/2k$.
4. At least half of the edges leaving each vertex are self-loops.
5. For every vertex v reachable from s, $\pi'(v) \geq 1/(2kD')$
6. For every vertex v and every edge label $(i, i') \in [d] \times [D']$, the (i, i')'th neighbor of v in G' equals the i'th neighbor of v in G, unless $v = s$ and $(i, i') \in B'$ where $B' \subseteq [d] \times [D']$ is a set of labels of density at most $n/D' \leq \varepsilon$.

Step 2: Blow up G' to a nearly biregular graph G'' We blow up each vertex v of G' to a "cloud" C_v consisting of $N_v = \lceil \pi'(v)N \rceil$ vertices, for a sufficiently large $N = O(kD'/\varepsilon)$. By Property 5 of G', we have $N_v \in [\pi'(v) \cdot N, (1+\varepsilon) \cdot \pi'(v) \cdot N]$. The vertex set of G'' is $V'' = \bigcup_v C_v$ for a total of $N'' = \sum_v N_v \in [N, (1+\varepsilon) \cdot N]$ vertices. Every vertex in G'' has degree $d \cdot D' \cdot D''$, for a sufficiently large $D'' = O(N/\varepsilon)$. For $(i, i', i'') \in [d] \times [D'] \times [D'']$, the (i, i', i'')'th edge leaving any vertex in C_u goes to the $(i'' \bmod N_v)$'th vertex of C_v, where v is the (i, i')'th neighbor of u in G'.

We now argue that G'' is nearly biregular, in the sense that all of the indegrees are close to $d \cdot D' \cdot D''$. Consider any vertex $\hat{v}$ in cloud C_v. Each edge (u, v) in G' induces either $N_u \cdot \lfloor D''/N_v \rfloor$ or $N_u \cdot \lceil D''/N_v \rceil$ edges into $\hat{v}$. Note that $D''/N_v \geq D''/((1+\varepsilon)N) \geq 1/\varepsilon$, if we choose $D'' \geq (1+\varepsilon)N/\varepsilon$. So the indegree of $\hat{v}$ is at most

$$\begin{aligned}
\sum_{(u,v)\in E'} N_u \cdot \left(\frac{D''}{N_v} + 1\right) &\leq \sum_{(u,v)\in E'} N_u \cdot (1+\varepsilon) \cdot \frac{D''}{N_v} \\
&\leq \sum_{(u,v)\in E'} [(1+\varepsilon)\pi'(u)N] \cdot (1+\varepsilon) \cdot \frac{D''}{\pi'(v)N} \\
&= \frac{(1+\varepsilon)^2 D''}{\pi'(v)} \cdot \sum_{(u,v)\in E'} \pi'(u) \\
&= \frac{(1+\varepsilon)^2 D''}{\pi'(v)} \cdot (d \cdot D' \cdot \pi'(v)) \\
&= (1+\varepsilon)^2 \cdot d \cdot D' \cdot D'' = (1 + O(\varepsilon)) \cdot d \cdot D' \cdot D''
\end{aligned}$$

By Property 3 of G', we observe that $|C_s| \geq \pi'(s)N \geq (1+\varepsilon) \cdot N''/2k$ and similarly $|C_t| \geq (1+\varepsilon) \cdot N''/2k$.

We now enumerate the properties of G'' established above.

1. Every vertex in G'' has out-degree $d \cdot D' \cdot D''$ and in-degree at most $(1 + O(\varepsilon)) \cdot d \cdot D' \cdot D''$.

2. For every vertex $\hat{u}$ in cloud C_u and every $(i, i', i'') \in [d] \times \times [D'] \times [D'']$, the (i, i', i'')'th edge leaving $\hat{u}$ leads to a vertex $\hat{v}$ in cloud C_v, where v is the (i, i')'th neighbor of u in G'. By Property 6 of G', v also equals the i'th neighbor of u in the original graph G unless $u = s$ and $(i, i', i'') \in B''$, where $B'' = B \times [D'']$ is a set of labels of density at most ε.

3. The number of edges between any two such vertices $\hat{u}$ and $\hat{v}$ equals either $e_{uv} \cdot \lfloor D''/N_v \rfloor$ or $e_{uv} \cdot \lceil D''/N_v \rceil$, where e_{uv} is the number of edges between u and v in G'.

4. C_s and C_t are both of density at least $1/2k$.

Step 3: Add edges to G'' to make a biregular graph G_{reg}. Property 1 of G'' implies that we can make the graph biregular by adding $O(\varepsilon \cdot d \cdot D' \cdot D'')$ edges leaving each vertex. Specifically, we obtain a biregular graph G_{reg} on the same vertex set as G'', in which every vertex has outdegree $d \cdot D_{\mathrm{reg}}$ for $D_{\mathrm{reg}} = (1+O(\varepsilon)) \cdot D' \cdot D''$. Each edge leaving a vertex has a label $(i, j) \in [d] \times [D_{\mathrm{reg}}]$, and the edges with $j \leq D' \cdot D''$ are identical to the edges of G''. We let $B_{\mathrm{reg}} = [d] \times ([D_{\mathrm{reg}}] \setminus [D' \cdot D''])$ be the set of remaining edge labels.

Let π_{reg} denote the uniform distribution on the set of vertices reachable from C_s. Since G_{reg} is biregular, this is a stationary distribution for G_{reg}. We now enumerate the properties of G''.

1. The vertex set of G_{reg} is $V_{\mathrm{reg}} = \bigcup_{v \in V} C_v$, and the outgoing edges are labelled by elements of $[d] \times [D_{\mathrm{reg}}]$

2. G_{reg} and G'' differ in at most $O(\varepsilon d D_{\mathrm{reg}})$ edges leaving and entering each vertex.

3. C_s and C_t are both of density at least $1/2k$.

4. There is a set $B \subseteq [d] \times [D_{\mathrm{reg}}]$ of density $O(\varepsilon)$ such that for every vertex $\hat{u} \in C_u$ and every edge label $(i, j) \in ([d] \times [D_{\mathrm{reg}}]) \setminus B$, the (i, j)'th neighbor of $\hat{u}$ in G_{reg} is in cloud C_v where v is the i'th neighbor of v in G. (Namely, take $B = B_{\mathrm{reg}} \cup B''$.)

5. For every $\hat{s} \in C_s$, we have $\lambda_{\hat{s}, \pi_{\mathrm{reg}}}(G_{\mathrm{reg}}) \leq 1 - 1/16\ell^2$.

All of these items follow from the previous discussion, except Property 5 bounding the expansion, which we proceed to do below.

Step 4: Analyze expansion of biregular graph. For this, it is useful to introduce a third Markov chain G''' on vertex set $V''' = V'' = V_{\mathrm{reg}}$, which is more closely related to random walks on G'. From any vertex $\hat{u} \in C_u$, the Markov chain G''' chooses a random neighbor v of u in G', and goes to a uniformly selected vertex $\hat{v} \in C_v$. It can be verified that the distribution π''' that assigns each vertex $\hat{v} \in C_v$ probability mass $\pi'''(\hat{v}) = \pi(v)/N_v$ is stationary for G'''. Moreover,

$$\lambda_{\hat{s}, \pi'''}(G''') = \lambda_{s, \pi'}(G') \leq 1 - \frac{1}{8\ell^2},$$

for any $\hat{s} \in C_s$.

We use this fact, and the fact that M''' is "close" to G_{reg} to bound $\lambda(G_{\mathrm{reg}})$. Specifically, let M_{reg}, M'', and M''' denote the transition matrices for G_{reg}, G'', and G''', respectively. Let $\rho = D_{\mathrm{reg}}/(D' \cdot D'') = 1 + O(\varepsilon)$ be the ratio between the degrees of G_{reg} and G''. We consider the two "error" matrices $\mathcal{E}_1 = \rho M_{\mathrm{reg}} - M''$, and $\mathcal{E}_2 = M'' - M'''$. To bound $\lambda_{\hat{s}, \pi_{\mathrm{reg}}}(G_{\mathrm{reg}})$, let x be any vector whose support is reachable from C_s such that $\langle x, \pi_{\mathrm{reg}} \rangle_{\pi_{\mathrm{reg}}} = 0$, i.e. $\sum_i x_i = 0$. We need to show that $\|M_{\mathrm{reg}} x\|_{\pi_{\mathrm{reg}}} \leq \lambda_{\mathrm{reg}} \cdot \|x\|_{\pi_{\mathrm{reg}}}$,

where $\lambda_{\text{reg}} = 1 - 1/16\ell^2$. Note that since π_{reg} is uniform, $\|\cdot\|_{\pi_{\text{reg}}}$ is simply a scaling of the standard Euclidean norm. We bound $\|M_{\text{reg}}x\|_{\pi_{\text{reg}}}$ as follows.

$$\|M_{\text{reg}}x\|_{\pi_{\text{reg}}} \leq \|\rho M_{\text{reg}}x\|_{\pi_{\text{reg}}} \leq \|M'''x\|_{\pi_{\text{reg}}} + \|\mathcal{E}_1 x\|_{\pi_{\text{reg}}} + \|\mathcal{E}_2 x\|_{\pi_{\text{reg}}}.$$

We bound each term separately. To bound the first, we first observe that the norms $\|\cdot\|_{\pi_{\text{reg}}}$ and $\|\cdot\|_{\pi'''}$ differ by a factor of at most $(1+\varepsilon)$, because π_{reg} and π''' almost identical. Specifically, for every vertex $\hat{v} \in C_v$, we have $\pi'''(v) = \pi'(v)/N_v$, $\pi_{\text{reg}}(v) = 1/N_{\text{reg}}$. These two quantities can be related as follows.

$$\frac{\pi'(v)}{N_v} \geq \frac{1}{(1+\varepsilon)N} \geq \frac{1}{(1+\varepsilon)N_{\text{reg}}}$$

and

$$\frac{\pi'(v)}{N_v} \leq \frac{1}{N} \leq \frac{1+\varepsilon}{N_{\text{reg}}}.$$

Thus, $\pi_{\text{reg}}(\hat{v}) \leq (1+\varepsilon)\cdot\pi'(\hat{v})$ and $\pi'(\hat{v}) \leq (1+\varepsilon)\cdot\pi_{\text{reg}}(\hat{v})$. This implies that the corresponding norms differ by a factor of at most $(1+\varepsilon)$. Therefore,

$$\begin{aligned}
\|M'''x\|_{\pi_{\text{reg}}} &\leq (1+\varepsilon)\cdot\|M'''x\|_{\pi'''} \\
&\leq (1+\varepsilon)\cdot\left(1 - \frac{1}{8\ell^2}\right)\cdot\|x\|_{\pi'''} \\
&\leq (1+\varepsilon)^2\cdot\left(1 - \frac{1}{8\ell^2}\right)\cdot\|x\|_{\pi_{\text{reg}}}.
\end{aligned}$$

For the second term, involving $\mathcal{E}_1$, we note that $\mathcal{E}_1$ equals $1/(dD'D'')$ times the adjacency matrix A of $G_{\text{reg}} \setminus G''$. Every vertex in this graph has outdegree $dD_{\text{reg}} - dD'D'' = \Theta(\varepsilon dD'D'')$, and indegree at most $O(\varepsilon dD'D'')$ (by Property 2). This implies that $\|Ax\|_{\pi_{\text{reg}}} \leq O(\varepsilon dD'D'')\cdot\|x\|_{\pi_{\text{reg}}}$. (One way to see this is to consider the the vector y assigning each *edge* (u,v) in $G_{\text{reg}} \setminus G''$, the value x_u. The squared length of y equals the squared length of x times the outdegree $\Theta(\varepsilon dD'D'')$. Then we obtain Ax by summing the entries of y incoming at each vertex. By Cauchy-Schwartz, this increases squared length by at most the maximum indegree $O(\varepsilon dD'D'')$.) Therefore, we have

$$\|\mathcal{E}_1 x\|_{\pi_{\text{reg}}} =\leq \frac{1}{dD'D''}\cdot\|Ax\|_{\pi_{\text{reg}}} = O(\varepsilon)\cdot\|x\|_{\pi_{\text{reg}}}.$$

Finally, we consider the third term, involving $\mathcal{E}_2$. We argue that each entry of $\mathcal{E}_2 = M'' - M'''$ is small. For vertices $\hat{u} \in C_u$, $\hat{v} \in C_v$, the $(\hat{u},\hat{v})$'th entry of M''' equals $(e_{uv}/(dD'))\cdot(1/N_v)$, by definition of G''', where e_{uv} is the number of edges between u and v in G'. On the other hand, by Property 3 of G'', the $(\hat{u},\hat{v})$'th entry of M'' is in the interval $[e_{uv}\cdot\lfloor D''/N_v\rfloor/(dD'D''), e_{uv}\cdot\lceil D''/N_v\rceil/(dD'D'')]$, which is contained in the interval $[e_{uv}/(dD'N_v) - 1/D'', e_{uv}/(dD'N_v) + 1/D'']$, since $e_{uv} \leq dD'$. Thus, each entry of $\mathcal{E}_2$ has absolute value at most $1/D''$. This implies that

$$\|\mathcal{E}_2 x\|_{\pi_{\text{reg}}} \leq \frac{\sqrt{N''}}{D''}\cdot\|x\|_{\pi_{\text{reg}}} \leq \varepsilon\cdot\|x\|_{\pi_{\text{reg}}},$$

where the last inequality comes by recalling that $N'' \leq (1+\varepsilon)\cdot N$ and $D'' \geq N/\varepsilon$.

Putting all of the above together, we have

$$\frac{\|M_{\text{reg}}x\|_{\pi_{\text{reg}}}}{\|x\|_{\pi_{\text{reg}}}} \leq (1+\varepsilon)^2\cdot\left(1 - \frac{1}{8\ell^2}\right) + O(\varepsilon) + \varepsilon \leq 1 - \frac{1}{16\ell^2},$$

provided $\varepsilon \geq c\cdot\ell^2$ for a sufficiently large constant c. ■

7 Combinatorial Measures

Other ways in which we can measure progress rather than spectral gaps are combinatorial measures such as edge expansion or vertex expansion.

Edge expansion is roughly preserved in the replacement product, but can deteriorate quite a bit when the graph is powered.

Theorem 7.1 *Let $G = (V, E)$ be a directed graph with n edges, such that every vertex has outdgree D_{out} and every indegree is at most D. Let ϵ be the edge expansion of G. Let H be a biregular directed graph with D vertices, degree d, and edge expansion δ. Then $G' := G \textcircled{r} H$ has edge expansion at least*

$$\frac{1}{4} \cdot \epsilon \cdot \frac{D_{\text{out}}}{D} \cdot \min\left\{\frac{1}{d+1}, \frac{\delta d}{d+1}\right\}$$

Concretely, we would use the replacement product using an inner graph H of constant degree and constant expansion, and D_{out} would be close to D in the outer graph, so that the expansion of $G \textcircled{r} H$ would be $\Omega(\epsilon)$.

Proof: [Of Theorem 7.1] Recall that, for a vertex v of G, the *cloud of* v is a set C_v of D vertices of G' that "correspond to" to v in the replacement product.

Let A be a set of less than $nD/2$ vertices of G'. We want to prove that there are at least

$$|A| \cdot d \cdot \left(\frac{1}{4} \cdot \epsilon \cdot \frac{D_{\text{out}}}{D} \cdot \min\left\{\frac{1}{d}, \delta\right\}\right)$$

edges from A to $\bar{A}$.

The intuition for the analysis is similar to the intuition in the analysis of the zig-zag graph product in [RVW]: if A is a disjoint union of clouds, then the expansion follows from the expansion of G, and if each cloud contains only a few elements of A then the expansion follows from the expansion of H. For a general set A, our analysis will use the expansion of G if most elements of A are concentrated in "half full" clouds; our analysis will use the expansion of G if most elements of A belong to "half empty" clouds.

Let $B \subseteq A$ be the subset of vertices of A that belong to "half-empty" clouds. That is, a vertex $w \in A$ is in B if it belongs to a cloud C_v such that at most $D/2$ elements of C_v are in A. For an half-empty cloud C_v, define $a_v = |A \cap C_v|$.

We consider the following two cases.

1. If $B > |A|\epsilon D_{\text{out}}/4D$, then each cloud C_v, $v \in S$, contributes at least $a_v \cdot \delta \cdot d$ to the cut between A and $\bar{A}$. (Here we are using the expansion of H.) Overall, the number of edges in the cut is at least

$$\sum_{v \in S} a_v \delta d \geq |B|\delta d \geq |A|\epsilon\delta d D_{\text{out}}/4D$$

2. If $|B| \leq |A|\epsilon D_{\text{out}}/4D$, then let T be the set of vertices v of G such that the cloud C_v contains at least $D/2$ elements of A. (These are the "half-full" clouds.) Note that $|T| \geq (|A| - |B|)/D > |A|/2D$.

   Now we have to consider two sub-cases:

(a) If $|T| \leq 3n/4$, then we claim that are at least $|A|\epsilon D_{\text{out}}/2D$ edges in G from T to $\bar{T}$. We prove the claim using the expansion of G. If $|T| \leq n/2$, then the number of edges from T to $\bar{T}$ is at least $|T|\epsilon D_{\text{out}} \geq |A|\epsilon D_{\text{out}}/2D$. If $n/2 \leq |T| \leq 3n/4$, then the number of edges from T to $\bar{T}$ is at least $|\bar{T}|\epsilon D_{\text{out}} \geq n\epsilon D_{\text{out}}/4 \geq |A|\epsilon D_{\text{out}}/2D$.

Those edges correspond to edges in G' that go from a vertex in a half-full cloud to a vertex in a half-empty cloud. We will argue that a reasonable fraction of such edges actually go from vertices in A to vertices in $\bar{A}$.

We first note that there are at most $|B| \leq |A|\epsilon D_{\text{out}}/4D$ edges in G' going to vertices in A that belong to half-empty clouds. Therefore, there are at least $|A|\epsilon D_{\text{out}}/4D$ edges in G' that have their first endpoint in a half-full cloud and their second endpoint in $\bar{A}$.

Let us now look at a half-full cloud C_v in G' from which there are, say, k_v outgoing edges whose second endpoint is a vertex in $\bar{A}$ in another cloud, and call $c_v = |C_v - A|$. We note that the cloud contributes at least $(k_v - c_v) + \delta d c_v \geq k_v \min\{1, \delta d\}$ edges to the cut between A and $\bar{A}$. This is because, of the k_v edges leaving C_v and going to a vertex in $\bar{A}$, at least $k_v - c_v$ originate from a vertex in A, and because the number of edges from $A \cap C_v$ to $C_v - A$ in C_v is at least $c_v \delta d$ because of the expansion of H.

Summing over all the clouds, we get a contribution that is at least

$$\sum_v k_v \min\{1, \delta d\} \geq |A|\epsilon \min\{1, \delta d\} D_{\text{out}}/4D$$

(b) If $|T| \geq 3n/4$, then we have $3n/4$ or more half-full clouds, each one containing between $D/2$ and D elements of A, even though $|A| \leq nD/2$. This means that of the $|T|$ half-full clouds, at least $n/2$ must contain at most $3D/4$ elements of A. (If we let c be the number of half-full clouds with at most $3D/4$ elements of A, we get $nD/2 \geq |A| \geq c \cdot D/2 + (|T| - c) \cdot 3D/4$, which, together with $|T| \geq 3n/4$, simplifies to $c \geq n/2$.) In each such cloud, the number of edges between A and $\bar{A}$ is at least $D\delta d/4$, so that the total number of edges between A and $\bar{A}$ is at least $nDd\delta/8$, which is at least $|A|d\delta/4$.

■

For directed graphs, as can be seen by the following example, the edge expansion does not necessarily improve by powering.

Proposition 7.2 *There is a directed graph G such that for every constant $t > 1$, the edge expansion of G^t is no better than that of G:*

$$\varepsilon(G^t) \leq \varepsilon(G)$$

Proof: We describe an unlabeled graph G because the labels are irrelevant in our case. Let G be the directed path on vertices $\{1, \ldots, 2n\}$ together with an additional edge from every vertex to 1. Formally, the edges of G are $(i, i+1)$ for all $i < 2n$ and also $(i, 1)$ for all i. To make the outdegree 2 everywhere duplicate the edge $(2n, 1)$. The edge expansion of this graph is obtained on the set $A = \{1, \ldots, n\}$. There is exactly one edge leaving this set in G, and since G is strongly connected the edge expansion is $\frac{E(A,\bar{A})}{2|A|} = 1/2n$.

The number edges leaving A in G^t is the number of length-t paths leaving A in G. For $t < n$, this number is equal to t. Since the out-degree of G^t is 2^t, the edge expansion of G (being the minimum over all choices of A) is bounded by $\frac{E(A,\bar{A})}{2^t|A|} = \frac{t}{2^t \cdot n} \leq 1/2n$.

Note that G can easily be made to have bounded in-degree, by 'spreading' the edges pointing to 1 to point somewhere among the first say $n/2$ vertices. ■

Acknowledgments

We are grateful to Irit Dinur for her invaluable collaboration during the early stages of this work and for her contribution to the results of Section 7. We also thank David Zuckerman for helpful discussions.

References

[AKS] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic Simulation in LOGSPACE. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 132–140, New York City, 25–27 May 1987.

[AKL$^+$] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, 29–31 Oct. 1979. IEEE.

[AS] N. Alon and B. Sudakov. Bipartite Subgraphs And The Smallest Eigenvalue. *Combinatorics, Probability & Computing*, 9(1), 2000.

[ATSWZ] R. Armoni, A. Ta-Shma, A. Wigderson, and S. Zhou. An $O(log(n)^{4/3})$ space algorithm for (s,t) connectivity in undirected graphs. *Journal of the ACM*, 47(2):294–311, 2000.

[BNS] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, pages 204–232, 15–17 May 1989.

[CPS] S. Caracciolo, A. Pelissetto, and A. Sokal. Two Remarks on Simulated Tempering. Unpublished manuscript (see [MR1]), 1992.

[Fil] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible markov chains with an application to the exclusion process. *Annals of Applied Probability*, 1:62–87, 1991.

[HW] S. Hoory and A. Wigderson. Universal Traversal Sequences for Expander Graphs. *Inf. Process. Lett.*, 46(2):67–69, 1993.

[MR1] N. Madras and D. Randall. Markov chain decomposition for convergence rate analysis. *Annals of Applied Probability*, 12:581–606, 2002.

[MR2] R. A. Martin and D. Randall. Sampling Adsorbing Staircase Walks Using a New Markov Chain Decomposition Method. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 492–502, Redondo Beach, CA, 17–19 Oct. 2000. IEEE.

[Mih] M. Mihail. Conductance and convergence of markov chains: a combinatorial treatment of expanders. In *In Proc. of the 37th Conf. on Foundations of Computer Science*, pages 526–531, 1989.

[Nis1] Nisan. RL $\subseteq$ SC. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 619–623, 1992.

[Nis2] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[NSW] N. Nisan, E. Szemeredi, and A. Wigderson. Undirected connectivity in $O(log^{1.5}n)$ space. In *Proceedings of the 30th FOCS*, pages 24–29, Research Triangle Park, North Carolina, 30 Oct.–1 Nov. 1989. IEEE.

[Rei] O. Reingold. Undirected ST-Connectivity in Log-Space. Technical Report TR04-94, Electronic Colloquium on Computational Complexity, November 1998. Extended abstract to appear in *STOC '05*.

[RVW] O. Reingold, S. Vadhan, and A. Wigderson. Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders. *Annals of Mathematics*, 155(1), January 2001. Extended abstract in *FOCS '00*.

[Sak] M. Saks. Randomization and Derandomization in Space-Bounded Computation. In *IEEE 11th Annual Conference on Structure in Complexity Theory*, 1996.

[SZ] M. Saks and S. Zhou. $\mathrm{BP_HSPACE}(S) \subseteq \mathrm{DSPACE}(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999. 36th IEEE Symposium on the Foundations of Computer Science (Milwaukee, WI, 1995).

[Sav] J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[SJ] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Inform. and Comput.*, 82(1):93–133, 1989.

[Wig] A. Wigderson. The complexity of graph connectivity. In *In Proceedings of the 17th Mathematical Foundations of Computer Science*, pages 112–132, 1992.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'