**CS225: Pseudorandomness** Prof. Salil Vadhan

Lecture 9 (revised): Explicit Constructions of Expanders and UNDIRECTED S-T CONNECTIVITY in **L**

*March 6 & 8, 2007*

Based on scribe notes by Adi Akavia and Vitaly Feldman.

# 1 Explicit Constructions

As discussed in previous lectures and the homework, expander graphs have numerous applications in theoretical computer science. For some of these applications, it may be acceptable to simply choose the graph at random, as we know that a random graph will be a good expander with high probability. For many applications, however, this simple approach does not suffice. Some reasons (in increasing order of significance):

- We may wish to tolerate the error probability introduced by the unlikely event that the graph is not an expander. To deal with this, we could try checking that the graph is an expander. Computing most combinatorial measures of expansion (e.g. vertex expansion or edge expansion) of a given graph is **NP**-hard, but the spectral expansion can be computed to high precision in time polynomial in the size of the graph (as it is just an eigenvalue computation). As we saw, spectral expansion does yield estimates on vertex expansion and edge expansion (albeit cannot give optimal expansion in these measures).
- Some of the applications of expanders (like the one from last time) are for reducing the amount of randomness needed for certain tasks, thus choosing the graph at random defeats the purpose.
- A number of the applications require *exponentially large* expander graphs, and thus we cannot even write down a randomly chosen expander. For example, for randomness-efficient error reduction of randomized algorithms, we need an expander on $2^m$ nodes where $m$ is the number of random bits used by the algorithm.

From a more philosophical perspective, finding explicit constructions is a way of developing and measuring our understanding of these fundamental combinatorial objects.

There are a couple of alternatives for defining explicit constructions of expanders on $N$ nodes are as follows.

**Mildly Explicit** Construction the complete representation of the graph in time poly$(N)$.

**Fully Explicit** Given a node $u \in [N]$ and $i \in [D]$ for $D$ the degree of the expander, compute the $i^{th}$ neighbor of $u$ in time poly$(\log N)$.

Consider the randomness-efficient error reduction application discussed in the last lecture, in which we performed a random walk on an expander graph with exponentially many nodes. Mild explicitness — though very straightforward — is insufficient for this application, as the desired expander graph is of exponential size, and hence cannot be even entirely stored, let alone constructed. But full explicitness is perfectly suited for efficiently conducting a random walk on a huge graph. We remark that, when the degree of the expander graph $D = \text{poly}(\log N)$, the third definition implies the second one. Thus the third definition is the one we adopt in this course.

**Goal**: Devise a fully explicit construction of a infinite family $\{G_i\}$ of $D$-regular $\lambda$ spectral expanders, where $D$ and $\lambda < 1$ are constants independent of $i$.

We remark that we would also like the set $\{N_i\}$, where $N_i$ is the number of vertices in $G_i$, to be not too sparse, so that the family of graphs $\{G_i\}$ has graphs of size close to any desired size.

# 2 Some Known Explicit Constructions

Here we mention a few known explicit constructions that are of interest because of their simple description, the parameters achieved, and/or the mathematics that goes into their analysis. We will not prove the expansion properties of any of these constructions (but will rather give a different explicit construction in the subsequent sections).

## 2.1 Gabber-Galil

The *Gabber-Galil Expander* is the graph $G = (V, E)$ with nodes $V = [M]^2$ (namely, each node is a pair of numbers from $\{0, ..., M-1\}$), and edges connecting each node $(x, y)$ with each of the following nodes: $(x, y), (x, x+y), (x, x+y+1), (x+y, y), (x+y, y)$ and $(x+y+1, y)$, where all arithmetic is mod $M$.

This is a fully explicit 5-regular graph with $N = M^2$ nodes and spectral expansion $\lambda < 1$, proved using Fourier analysis. We note that this graph is bipartite (equivalently, directed), however, it can be transformed to a non-bipartite (as you will show on Problem Set 3).

## 2.2 Cycle mod $p$ with inverse chords

The graph $G = (V, E)$ with vertex set $V = \mathbb{F}_p$ the finite field of prime order $p$ and edges that connect each node $x$ with the nodes: $x+1$, $x-1$ and $x^{-1}$ (where all arithmetic is mod $p$ and we define $0^{-1}$ to be 0).

This graph is only mildly explicit since we do not know how to construct $n$-bit primes deterministically in time $\text{poly}(n)$ (though it is conjectured that we can do so by simply checking the first $\text{poly}(n)$ $n$-bit numbers). The proof of expansion relies on a deep theorem in number theory (the Selberg 3/16 Theorem).

## 2.3 Ramanujan Graphs

The graph $G = (V, E)$ with nodes $V = \mathbb{F}_q \cup \{\infty\}$ elements in the finite field of prime order $q$ s.t. $q \equiv 1 \bmod 4$ plus one extra node representing infinity. The edges in this graph connect each node $z$ with all

$$z' = \frac{(a_0 + ia_1)z + (a_2 + ia_3)}{(-a_2 + ia_3)z + (a_0 - ia_1)}$$

for $a_0, a_1, a_2, a_3 \in \mathbb{N}$ s.t. $a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$ such that $a_0 > 0$ and $a_1, a_2, a_3$ are even, for some fixed prime $p \neq q$ satisfying $p \equiv 1 \bmod 4$ and where $i^2 = -1 \bmod q$.

The degree of the graph turns out to be $D = p + 1$, as there are $p + 1$ solutions to the equation $a_0^2+a_1^2+a_2^2+a_3^2 = p$, and it is an *optimal* spectral expander because $\lambda(G) \leq 2\sqrt{D-1}/D$ (even better than we know for random graphs, which have an additive $o(1)$ term in the spectral expansion). These graphs are also only mildly explicit.

These are called Ramanujan graphs because their spectral expansion relies on deep results in number theory concerning the Ramanujan Conjectures. Subsequently, the term *Ramanujan graphs* came to refer to any infinite family of graphs with optimal spectral expansion $2\sqrt{D-1}/D$.

# 3 Graph Operations

We will not prove the expansion of any of the above constructions. Instead we will present an iterative construction, where we start with a "constant size" expander $H$ and repeatedly applying graph operations to get bigger expanders. The operations that we apply should increase the number of nodes in the graph, while keeping the degree and the spectral expansion $\lambda$ bounded. We'll see three operations, each improving one property while paying a price on the others; however, combined together, they yield the desired expander.

The benefit of this approach is that each of the operations have simple combinatorial intuition, and they will also be useful in derandomizing the logspace algorithm for Undirected S-T Connectivity.

We use the notation $(N, D, \lambda)$-*graph* to denote a $D$-regular graph $\lambda$ spectral expander on $N$ nodes.

## 3.1 Squaring

**Definition 1 (Squaring Operation $G^2$)** *If $G = (V, E)$ is a $D$-regular graph, then $G^2 = (V, E')$ is a $D^2$-regular graph on the same vertex set, where the $(i, j)$'th neighbor of a vertex $x$ is the $j$'th neighbor of the $i$'th neighbor of $x$. In particular, a random step on $G^2$ consists of two random steps on $G$.*

**Lemma 2** *If $G$ is a $(N, D, \lambda)$-graph, then $G^2$ is a $(N, D^2, \lambda^2)$-graph. Namely, the degree deteriorates by squaring, while the spectral expansion is improved.*

**Proof:** The affect of squaring on the number of nodes $N$ and the degree $D$ is immediate from the definition. The spectral expansion $\lambda$ is squared, since if $M$ is the random-walk matrix for $G$, then $M^2$ is the random-walk matrix for $G^2$. So for any vector $x \perp u$,

$$\|xM^2\| \leq \lambda \cdot \|xM\| \leq \lambda^2 \|x\|.$$

■

### 3.2 Tensoring

**Definition 3 (Tensor Operation $G_1 \otimes G_2$)** *Let $G_1 = (V_1, E_1)$ be $D_1$-regular and $G_2 = (V_2, E_2)$ be $D_2$-regular. Then their* tensor product *is the $D_1D_2$-regular graph $G_1 \otimes G_2 = (V_1 \times V_2, E)$, where the $(i_1, i_2)$'th neighbor of a vertex $(x_1, x_2)$ is $(y_1, y_2)$, where $y_b$ is the $i_b$'th neighbor of $x_b$ in $G_b$. That is, a random step on $G_1 \otimes G_2$ consists of a random step on $G_1$ in the first component and a random step on $G_2$ in the second component.*

**Definition 4 (Tensor Operation on Vectors)** *Let $x \in \mathbb{R}^{N_1}, y \in \mathbb{R}^{N_2}$, then their* tensor product *$z = x \otimes y \in \mathbb{R}^{N_1N_2}$ is the vector[1] $z_{ij} = x_i y_j$.*

*Similarly, for matrices $A = (a_{ij}) \in \mathbb{R}_{N_1 \times N_1}, B = (b_{ij}) \in \mathbb{R}_{N_2 \times N_2}$, their* tensor product *$C = A \otimes B \in \mathbb{R}_{N_1N_2 \times N_1N_2}$ is the matrix $C = (c_{ij})$ where $c_{ij,i'j'} = a_{ii'}b_{jj'}$.*

A few comments on the tensor operation:

- A random walk on a tensor graph $G_1 \otimes G_2$ is equivalent to taking two independent random walks on $G_1$ and $G_2$
- For vectors $x \in \mathbb{R}^{N_1}, y \in \mathbb{R}^{N_2}$ that are probability distributions (namely, normalized non-negative vectors), their tensor product $x \otimes y$ is a probability distribution on $[N_1] \times [N_2]$ elements where the two components are independently distributed according to $x$ and $y$, respectively.
- $(x \otimes y)(A \otimes B) = (xA) \otimes (yB)$, and in fact $A \otimes B$ is the unique matrix with this property.
- Not all vectors $z \in \mathbb{R}^{N_1N_2}$ are a tensor product of two vectors $x \in \mathbb{R}^{N_1}, y \in \mathbb{R}^{N_2}$; nonetheless, the set of all vectors $z$ that are a tensor product of some two such vectors spans $\mathbb{R}^{N_1N_2}$.
- If $M_1, M_2$ are the random-walk matrices for graphs $G_1, G_2$ respectively, then $M_1 \otimes M_2$ is the random-walk matrix for the graph $G_1 \otimes G_2$ is

  $$M_1 \otimes M_2 = (I_{N_1} \otimes M_2)(M_1 \otimes I_{N_2}) = (M_1 \otimes I_{N_2})(I_{N_1} \otimes M_2),$$

  where $I_N$ denotes the $N \times N$ identity matrix. That is, we can view a random step on $G_1 \otimes G_2$ as being a random step on the $G_1$ component followed by one on the $G_2$ component or vice-versa.

**Lemma 5** *If $G_1$ is an $(N_1, D_1, \lambda_1)$-graph and $G_2$ is an $(N_2, D_2, \lambda_2)$-graph, then $G_1 \otimes G_2$ is an $(N_1N_2, D_1D_2, \max\{\lambda_1, \lambda_2\}$ graph. In particular, if $G_1 = G_2$, then the number of nodes improves, the degree deteriorates, and the spectral expansion remains unchanged.*

**Proof:** The intuition for the construction is as follows. Any probability distribution on the vertices $(v_1, v_2)$ of $G_1 \otimes G_2$ can be thought of as picking a cloud $v_1$ according to some marginal distribution and then picking the vertex $v_2$ within the cloud $v_2$ according to some conditional distribution given

[1] For convenience we index the vector $z$ by two indices $i, j$. To transform to a standard indexing we can map $ij$ to $iN_1 + j$.

$v_1$. If the overall distribution on pairs is far from uniform, then either the marginal distribution on the clouds must be far from uniform or the conditional distribution within the clouds must be far from uniform. When we take a random step, the expansion of $G_1$ will bring us closer to uniform in the first case and the expansion of $G_2$ will bring us closer to uniform in the second.

One way to actually prove the bound is to use the fact that the eigenvalues of $M_1 \otimes M_2$ are all the products of eigenvalues of $M_1$ and $M_2$ (so the three largest absolute values are $\{1 \cdot 1, \lambda_1 \cdot 1, 1 \cdot \lambda_2\}$), but we instead use the vector decomposition method to give a proof that matches the intuition more closely. Given any vector $x \in \mathbb{R}^{N_1 N_2}$ that is orthogonal to $u_{N_1 N_2}$, we can decompose $x$ as $x = x^{\parallel} + x^{\perp}$ where $x^{\parallel}$ is a multiple of $u_{N_2}$ on each cloud of size $N_2$ and $x^{\perp}$ is orthogonal to $u_{N_2}$ on each cloud. Note that $x^{\parallel} = y \otimes u_{N_2}$, where $y \in \mathbb{R}^{N_1}$ is orthogonal to $u_{N_1}$ (because $x^{\parallel} = x - x^{\perp}$ is orthogonal to $u_{N_1 N_2}$). If we think of $x$ as the nonuniform component of a probability distribution, then $x^{\parallel}$ and $x^{\perp}$ correspond to the two cases in the intuition above.

For the first case, we have

$$x^{\parallel} M = (y \otimes u_{N_2})(M_1 \otimes M_2) = (yM_1) \otimes u_{N_2}.$$

The expansion of $G_1$ tells us that $M_1$ shrinks $y$ by a factor of $\lambda_1$, and thus $M$ shrinks $x^{\parallel}$ by the same factor. For the second case, we write

$$x^{\perp} M = x^{\perp}(I_{N_1} \otimes M_2)(M_1 \otimes I_{N_2}).$$

The expansion of $G_2$ tells us that $M_2$ will shrink $x^{\perp}$ by a factor of $\lambda_2$ on each cloud, and thus $I_{N_1} \otimes M_2$ will shrink $x^{\perp}$ by the same factor. The subsequent application of $M_1 \otimes I_{N_2}$ cannot increase the length (being the random-walk matrix for a regular graph, albeit a disconnected one). Thus, $\|x^{\perp} M\| \leq \lambda_2 \|x^{\perp}\|$.

Finally, we argue that $x^{\parallel} M$ and $x^{\perp} M$ are orthogonal. Note that $x^{\parallel} M = (yM_1) \otimes u_{N_2}$ is a multiple of $u_{N_2}$ on every cloud. Thus it suffices to argue that $x^{\perp}$ remains orthogonal to $u_{N_2}$ on every cloud after we apply $M$. Applying $(I_{N_1} \otimes M_2)$ retains this property (because applying $M_2$ preserves orthogonality to $u_{N_2}$, by regularity of $G_2$) and applying $(M_1 \otimes I_{N_2})$ retains this property because it assigns each cloud a linear combination of several other clouds (and a linear combination of vectors orthogonal to $u_{N_2}$ is also orthogonal to $u_{N_2}$).

Thus,

$$\begin{aligned}
\|xM\|^2 &= \|x^{\parallel} M\|^2 + \|x^{\perp} M\|^2 \\
&\leq \lambda_1^2 \|x^{\parallel}\|^2 + \lambda_2^2 \|x^{\perp}\|^2 \\
&\leq \max\{\lambda_1, \lambda_2\}^2 \cdot (\|x^{\parallel}\|^2 + \|x^{\perp}\|^2) \\
&= \max\{\lambda_1, \lambda_2\}^2 \cdot \|x\|^2,
\end{aligned}$$

as desired. ■

### 3.3 The Zig-Zag Product

Let $G$ be an $(N_1, D_1, \lambda_1)$ expander and $H$ be a $(D_1, D_2, \lambda_2)$ expander. The *zig-zag product* of $G$ and $H$, denoted $G ⓩ H$, will be defined according to the following intuition. The nodes of $G ⓩ H$ are the pairs $(u, i)$ where $u \in V(G)$ and $i \in V(H)$. We are interested in performing a random walk on a graph $G$ using a random walk on $H$ to choose the edge in $G$ (notice that this is the reason why we require the number of vertices in $H$ to be equal to the degree of $G$). A step in $G \textcircled{z} H$ can therefore be viewed as a step to a random neighbor in $H$ and then a step in $G$ to a neighbor whose index is equal to the label of the current node in $H$. This idea is very sensible since a random walk on an a "good" expander graph $H$ should generate choices that are sufficiently random to produce a random walk on $G$. One problem with this definition is that it is not symmetrical. That is, the fact that you can go from $(u, i)$ to $(v, j)$ does not mean that you can go from $(v, j)$ to $(u, i)$. We correct this by adding another step in $H$ after the step in $G$.

More formally,

**Definition 6 (Zig-zag Product)** *$G \textcircled{z} H$ is a graph whose vertices are pairs $(u, i)$ where $u \in V(G)$ and $i \in V(H)$. The $(a, b)$'th neighbor of a vertex $(u, i)$ is the vertex $(v, j)$ computed as follows:*

1. *Let $i'$ be the $a$'th neighbor of $i$ in $H$.*
2. *Let $v$ be the $i'$'th neighbor of $u$ in G, and let $j'$ be such that $u$ is the $j'$'th neighbor of $v$ in G.*
3. *Let $j$ be the $b$'th neighbor of $j'$ in $H$.*

**Theorem 7** *If $G$ is a $(N_1, D_1, \lambda_1 = 1 - \gamma_1)$-graph, and $H$ is a $(D_1, D_2, \lambda_2 = 1 - \gamma_2)$-graph then $G \textcircled{z} H$ is a $(N_1 D_1, D_2^2, \lambda = 1 - \gamma_1 \cdot \gamma_2^2)$-graph. In particular, $\lambda \leq \lambda_1 + 2\lambda_2$.*

The graphs $G$ should be thought of as a big graph and $H$ a small graph, where $D_1$ is a large constant and $D_2$ is a small constant. Note that the number of nodes in $H$ is required to equal the degree of $G$. Observe that when $D_1 > D_2^2$ the degree is reduced by the Zig-Zag product. As for the effect on the spectral expansion, though the bound stated in the proposition suffices to our needs, we comment that actually, a stronger bound holds; in particular $\lambda \leq \lambda_1 + \lambda_2$.

There are two different intuitions underlying the expansion of the zig-zag product:

- Given an initial distribution on the vertices of $G_1 \textcircled{z} G_2$ that is far from uniform, there are two extreme cases. Either (a) all the distributions within the clouds are far from uniform, or (b) all the distributions within the 'clouds' of size $D_1$ are uniform (in which case the marginal distribution on the clouds must be far from uniform). In case (a), the first $H$-step already brings us closer to the uniform distribution, and the other two steps cannot hurt (as they are steps on regular graphs). In case (b), the first $H$-step has no effect, but the $G$-step has the effect of making the marginal distribution on clouds closer to uniform. But note that we haven't actually gotten closer to the uniform distribution on the vertices of $G_1 \textcircled{z} G_2$ because the $G$-step is a permutation. Still, if the marginal distribution on clouds has become closer to uniform, then the distributions within the clouds must have become further from uniform, and thus the second $H$-step brings us closer to uniform. This leads to a proof by *vector decomposition*, where we decompose any vector $x$ that is orthogonal to uniform into components $x^{\parallel}$ and $x^{\perp}$, where $x^{\parallel}$ is uniform on each cloud, and $x^{\perp}$ is orthogonal to uniform on each cloud. This approach gives the best known bounds on the spectral expansion of the zig-zag product, but it can be a bit messy since the two components do not necessarily remain orthogonal (unlike the case of the tensor product, where we were able to show that $x^{\parallel} M$ is orthogonal to $x^{\perp} M$).

- The second intuition is to think of the expander $H$ as behaving 'similarly' to the complete graph on $D_1$ vertices (with self-loops). In the case that $H$ equals the complete graph, then it is easy to see that $G \text{ⓩ} H = G \otimes H$. Thus it is natural to apply *matrix decomposition*, writing the random-walk matrix for an arbitrary expander $H$ as a convex combination of the random-walk matrix for the complete graph and an error matrix. This gives a very clean analysis, but not the best known bounds.

We now proceed with the formal proof.

**Proof of Theorem 7:** Let $M$ be the random-walk matrix for $G_1 \text{ⓩ} G_2$. We decompose $M$ into the product of three matrices, corresponding to the three steps in the definition of $G_1 \text{ⓩ} G_2$'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 $\hat{A}$ be the permutation matrix corresponding to the $G_1$-step. That is, $\hat{A}_{(u,i),(v,j)}$ is 1 iff $v$ is the $i$'th neighbor of $u$ and $u$ is the $j$'th neighbor of $v$. By the definition of $G_1 \text{ⓩ} G_2$, we have $M = \tilde{B}\hat{A}\tilde{B}$.

By the Matrix Decomposition Lemma, $B = \gamma_2 J + (1-\gamma_2)E$, where every entry of $J$ equals $1/D_1$ and $E$ has norm at most 1. Then $\tilde{B} = \gamma_2 \tilde{J} + (1-\gamma_2)\tilde{E}$, where $\tilde{J} = I_{N_1} \otimes J$ and $\tilde{E} = I_{N_1} \otimes E$ has norm at most 1.

This gives

$$M = \left(\gamma_2 \tilde{J} + (1-\gamma_2)\tilde{E}\right) \hat{A} \left(\gamma_2 \tilde{J} + (1-\gamma_2)\tilde{E}\right) = \gamma_2^2 \tilde{J}\hat{A}\tilde{J} + (1-\gamma_2^2)F,$$

where $F$ has norm at most 1. Now, the key observation is that $\tilde{J}\hat{A}\tilde{J} = A \otimes J$.

Thus,

$$M = \gamma_2^2 \cdot A \otimes J + (1-\gamma_2^2)F,$$

and thus

$$\begin{aligned} \lambda(M) &\leq \gamma_2^2 \cdot \lambda(A \otimes J) + (1-\gamma_2^2) \\ &\leq \gamma_2^2 \cdot (1-\gamma_1) + (1-\gamma_2^2) \\ &= 1 - \gamma_1 \gamma_2^2, \end{aligned}$$

as desired. ■

# 4 The Expander Construction

**Mildly Explicit Construction.** As a first attempt for constructing a family of expanders, we construct an infinite family $G_1, G_2, \ldots$ of graphs utilizing only the squaring and the zig-zag operations: Let $H$ be a $(D^4, D, \lambda_0)$-graph for $\lambda_0 = 1/8$ (as you will construct on PS3).

$$\begin{aligned} G_1 &= H^2 \\ G_{t+1} &= G_t^2 \text{ⓩ} H \end{aligned}$$

**Proposition 8** *For all $t$, $G_t$ is a $(D^{4t}, D^2, 1/2)$-graph.*

**Proof:** We show by induction that $G_t$ is a $(D^{4t}, D^2, \lambda)$-graph for $\lambda = 1/2$. Induction base: by the definition of $H$, $G_1 = H^2$ is a $(D^4, D^2, \lambda_0^2)$-graph.

Checking the induction step, first note that $G_t^2 ⓩ H$ is well-defined, as by induction hypothesis, $G_t$ is of degree $D^2$ implying $G_t^2$ is of degree $D^4 = \#$nodes in $H$. Now,

$$\begin{aligned}
deg(G_{t+1}) &= deg(H)^2 = D^2 \\
\#nodes(G_{t+1}) &= \#nodes(G_t^2) \cdot \#node(H) = N_t \cdot D^4 = D^{4t} D^4 = D^{4(t+1)} \\
\lambda(G_{t+1}) &\leq \lambda^2 + 2\lambda_0 = (1/2)^2 + 2 \cdot (1/8) = 1/2
\end{aligned}$$

■

Now, we recursively bound the time to compute neighbors in $G_t$. Actually, due to the way the $G$-step in the zig-zag product is defined, we actually bound the time to compute the *edge-rotation map* $(u, i) \mapsto (v, j)$, where $v$ is the $i$'th neighbor of $u$ and $u$ is the $j$'th neighbor of $v$. Denote by time$(G_t)$ the time required for one evaluation of the edge-rotation map for $G_t$. This requires two evaluations of the rotation map for $G_{t-1}$ (the squaring requires two applications, while the zig-zag part requires only one). Therefore,

$$\begin{aligned}
\text{time}(G_t) &= 2 \cdot \text{time}(G_{t-1}) + \text{poly}(\log N_t) \\
&= 2^t \cdot \text{poly}(\log N_t) \\
&= N_t^{\Theta(1)},
\end{aligned}$$

where the last equality holds because $N_t = D^{4t}$ for a constant $D$). Thus, this construction is only mildly explicit.

**Fully explicit construction.** We remedy the above difficulty by using tensoring to make the sizes of the graphs grow more quickly. We construct an infinite family $G_1, G_2, ...$ of graphs utilizing the squaring, tensor and zig-zag operations: Let $H$ be a a $(D^8, D, 1/8)$-graph.

$$\begin{aligned}
G_1 &= H^2 \\
G_{t+1} &= (G_t \otimes G_t)^2 ⓩ H'
\end{aligned}$$

In this family of graphs, the number of nodes grow doubly exponentially $N_t \approx c^{2^t}$, while computation time grows only exponentially as before. Namely,

$$\text{time}(G_t) = 4^t \cdot \text{poly}(\log N_t) = \text{poly}(\log N_t).$$

We remark that the above family is rather sparse, namely, the numbers in $\{N_t\}$ are far apart. To overcome this shortcoming, we can amend the above definition to have

$$G_t = (G_{\lceil t/2 \rceil} \otimes G_{\lfloor t/2 \rfloor})^2 ⓩ H.$$

The density of this amended family of graphs is $D^{8t}$, so given a number $N$, we can find a graph $G_t$ in the family whose size is at most $D^8 \cdot N = O(N)$. It is not difficult to verify that the construction remains fully explicit.

## 5 UNDIRECTED S-T CONNECTIVITY in logspace

Recall the UNDIRECTED S-T CONNECTIVITY problem: given an undirected graph $G$ and two vertices $s, t$, decide whether there is a path from $s$ to $t$. In Lecture 4, we saw that this problem can be solved in randomized logspace (**RL**). Here we will see a very recent result (2005) how we can use expanders and the operations above to solve this problem in deterministic logspace (**L**).

The algorithm is based on the following two ideas:

- UNDIRECTED S-T CONNECTIVITY can be solved in logarithmic-space on constant-degree expander graphs. More precisely, it is easy on constant-degree graphs where every connected component is an expander (i.e. has spectral expansion bounded away from 1).
- The same operations we used to construct an infinite expander family above can also be used to turn *any* graph into an expander (in logarithmic space). Above, we started with a constant-sized expander and used the various operations to build larger and larger expanders. There, the goal was to increase the size of the graph (which was accomplished by tensoring and/or zig-zag), while not hurting the degree and the expansion (which was accomplished by zig-zag and squaring, which made up for losses in these parameters). Here, we want to improve the expansion (which will be accomplished by squaring), without losing in the degree (as will be handled by zig-zag) or making the graph too much bigger (so we will not use tensoring).

Specifically, the algorithm is as follows.

**Algorithm for** UNDIRECTED S-T CONNECTIVITY**.** On input a graph $G$ with $N$ edges and vertices $s$ and $t$:

1. Let $H$ be a fixed $(D^4, D, 1/4)$ graph for some constant $D$.
2. Reduce $(G, s, t)$ to $(G_0, s_0, t_0)$, where $G_0$ is a $D^2$-regular, nonbipartite graph and $s_0$ and $t_0$ are connected in $G_0$ iff $s$ and $t$ are connected in $G$.
3. For $k = 1, \ldots, \ell = O(\log N)$, define:
   (a) Let $G_k = G_{k-1}^2 ⓩ H$
   (b) Let $s_k$ and $t_k$ be vertices in the 'clouds' in $G_k$ corresponding to $s_{k-1}$ and $t_{k-1}$, respectively, so that $s_k$ and $t_k$ are connected in $G_k$ iff $s_{k-1}$ and $t_{k-1}$ are connected in $G_{k-1}$.
4. Try all paths of length $O(\log N)$ in $G_\ell$ from $s_\ell$ and accept if any of them visit $t_\ell$.

We will discuss how to implement this algorithm in logspace later, and first analyze its correctness. Let $C_k$ be the connected component of $G_k$ containing $s_k$. Observe that $C_k = C_{k-1}^2 ⓩ H$. As you showed on Problem Set 2 for any nonbipartite undirected graph, $\gamma(C_0) \geq 1/\text{poly}(N)$. We will argue that in each iteration the spectral gap increases by a constant factor, and thus after $O(\log N)$ iterations we have an expander.

By Lemma 2, we have $1 - \gamma(C_{k-1}^2) \leq (1 - \gamma(C_{k-1}))^2$. That is,

$$\gamma(C_{k-1}^2) \geq 2 \cdot \gamma(C_k) \cdot (1 - \gamma(C_k)) \approx 2\gamma(C_k)$$

for small $\gamma(C_k)$. By Theorem 7, we have

$$\begin{aligned}\gamma(C_{k-1}^2 ⓩ H) &\geq \gamma(H)^2 \cdot \gamma(C_{k-1}) \\ &= \left(\frac{3}{4}\right)^2 \cdot 2 \cdot \gamma(C_k) \cdot (1-\gamma(C_k)) \\ &\geq \min\left\{\frac{17}{16} \cdot \gamma(C_k), \frac{1}{18}\right\}\end{aligned}$$

where the last inequality is obtained by considering whether $\gamma(C_k) \leq 1/18$ or $\gamma(C_k) > 1/18$. Thus, after $\ell = O(\log N)$ iterations, we must have $\gamma(C_\ell) \geq 1/18$. Moreover, observe that the number of vertices $N_\ell$ in $G_\ell$ is at most $N_0 \cdot (D^4)^\ell = \mathrm{poly}(N)$, so considering paths of length $O(\log N)$ will suffice to decide $s$-$t$ connectivity in $G_\ell$.

To show that the algorithm can be implemented in logarithmic space, we argue that the rotation map of each $G_k$ can be computed with only $O(1)$ more space than the rotation map of $G_{k-1}$, so that $G_\ell$ requires space $O(\log N) + O(\ell) = O(\log N)$.

Formally, let $\mathrm{space}(G_k)$ denote the workspace needed to compute the edge-rotation map of $G_\ell$ in the following model:

- Input Description:
  - Tape 1 (read-only): Contains the initial input graph $G$, with the head at the leftmost position of the tape.
  - Tape 2 (read-write): Contains the input pair $(v, i)$ where $v$ is a vertex of $G_i$ and $i \in [D^2]$ is an index of the a neighbor on a *read-write* tape, with the head at the *rightmost* position of $i$. The rest of the tape may contain additional data.
  - Tapes 3+ (read-write): Blank worktapes with the head at the leftmost position.
- Output Description:
  - Tape 1: The head should be returned to the leftmost position.
  - Tape 2: In place of $(v, i)$, it should contain the output pair $(w, j)$ where $w$ is the $i$'th neighbor of $v$ and $v$ is the $j$'th neighbor of $w$. The head should be at the rightmost position of $j$ and the rest of the tape should remain unchanged.
  - Tapes 3+ (read-write): Are returned to blank state with the heads at the leftmost position.

In this model, it is not difficult to argue that $\mathrm{space}(G_0) = O(\log N)$, and $\mathrm{space}(G_k) = \mathrm{space}(G_{k-1}) + O(1)$. For the latter, we first argue that $\mathrm{space}(G_{k-1}^2) = \mathrm{space}(G_{k-1}) + O(1)$, and then that $\mathrm{space}(G_{k-1}^2 ⓩ H) = \mathrm{space}(G_{k-1}^2) + O(1)$. For $G_{k-1}^2$, we are given a triple $(v, (i_1, i_2))$, with the head on the rightmost position of $i_2$, and both $i_1$ and $i_2$ are elements of $[D^2]$ (and thus of constant size). We move the head left to the rightmost position of $i_1$, compute the edge-rotation map of $G_{k-1}$ on $(v, i_1)$ so that the tape contents are now $(w, j_1, i_2)$. Then we swap $j_1$ and $i_2$, and run the edge-rotation map of $G_{k-1}$ on $(w, i_2)$ to get $(w, j_2, j_1)$. For the $G_{k-1}^2 ⓩ H$, we are given a triple $((v, i), (a_1, a_2)$, where $v$ is a vertex of $G_{k-1}^2$, $i$ is a vertex of $H$ or edge-label for $G_{k-1}^2$, and $a_1, a_2$ are edge labels for $H$. Evaluating the rotation map requires two evaluations of the rotation map for $H$ (both of which are 'constant-size' operations) and one evaluation of the rotation map of $G_{k-1}^2$.

Thus we have proven:

**Theorem 9** UNDIRECTED S-T CONNECTIVITY *is in* $\mathbf{L}$.

We remark that proving $\mathbf{RL} = \mathbf{L}$ in general remains open. The best deterministic simulation known for $\mathbf{RL}$ is $\mathbf{L}^{3/2} = \mathbf{DSPACE}(\log^{3/2} n)$, which makes beautiful use of known pseudorandom generators for logspace computation. Unfortunately, we will not have time to cover this line of work this semester. Historically, improved derandomizations for UNDIRECTED S-T CONNECTIVITY have inspired improved derandomizations of $\mathbf{RL}$ (and vice-versa). Since Theorem 9 is still quite recent, there is a good chance that we have not yet exhausted the ideas in it.

**Open Problem 10** *Show that* $\mathbf{RL} \subseteq \mathbf{L}^c$ *for some* $c < 3/2$.

Another open question is the construction of *universal traversal sequences* — fixed walks of polynomial length that are guaranteed to visit all vertices in any connected undirected graph of a given size. Using the ideas from the algorithm above, it is possible to obtain logspace-constructible, polynomial-length universal traversal sequences for all regular graphs that are *consistently labelled* in the sense that it is never the case that the $i$'th neighbor of a vertex $v$ is equal to the $i$'th neighbor of a vertex $w \neq u$. For general labellings, the best known universal traversal sequences are of length $N^{O(\log N)}$ (and are constructible in space $O(\log^2 N)$).

**Open Problem 11** *Give an explicit construction of universal traversal sequences of polynomial length for arbitrarily labelled undirected graphs (or even for an arbitrary labelling of the complete graph!).*

We remark that handling general labellings (for an object related to universal traversal sequences) seems to be the main obstacle in extending Theorem 9 to prove $\mathbf{RL} = \mathbf{L}$.