The Zig-Zag Product and Reingold's Theorem

Anurag Sahay

University of Rochester

asahay@ur.rochester.edu

27th February, 2020

Overview of the Talk

References

Omer Reingold.
Undirected connectivity in log-space.
Journal of the ACM (JACM), 55(4):1–24, 2008.

Omer Reingold, Salil Vadhan, and Avi Wigderson.
Entropy waves, the zig-zag graph product, and new constant-degree expanders.
Annals of mathematics, 155(1):157–187, 2002.

Shlomo Hoory, Nathan Linial, and Avi Wigderson.
Expander graphs and their applications.
Bulletin of the American Mathematical Society, 43(4):439–561, 2006.

Luca Trevisan.
Lecture Notes – U.C. Berkeley,
CS278: Computational Complexity, 2004.
https://people.eecs.berkeley.edu/~luca/cs278-04/notes/.

`ustcon` and L

Recall the definition of `ustcon` and L:

Definition (Undirected *st*-connectivity)

`ustcon` is the following decision problem:

Input: an undirected graph $G = (V, E)$, and two vertices $s, t \in V$

Output: 1 if $s \rightsquigarrow t$ in G, and 0 otherwise.

ustcon and L

Recall the definition of ustcon and L:

Definition (Undirected *st*-connectivity)

ustcon is the following decision problem:

Input: an undirected graph $G = (V, E)$, and two vertices $s, t \in V$

Output: 1 if $s \rightsquigarrow t$ in G, and 0 otherwise.

Definition (LogSpace)

$$L = DSPACE(\log n)$$

In other words, L contains all languages $\mathtt{L} \subseteq \{0, 1\}^*$ for which a Turing machine M decides $\mathtt{L}$ using no more that $\mathcal{O}(\log n)$ space on input of length n.

Reingold's Theorem

The goal of this talk is to describe the proof of Reingold's theorem:

Theorem (Reingold, 2005)

$$\texttt{ustcon} \in L$$

Note that the graph is not assumed to be simple (multiedges and loops are both allowed). We will restrict ourselves largely to regular graphs, and then show how the problem in a general graph can be reduced to the regular case.

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph
- The graph satisfies a strong isoperimetric inequality

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph
- The graph satisfies a strong isoperimetric inequality
- Every set of vertices has many neighbours

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph
- The graph satisfies a strong isoperimetric inequality
- Every set of vertices has many neighbours
- Every cut has many edges across it

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph
- The graph satisfies a strong isoperimetric inequality
- Every set of vertices has many neighbours
- Every cut has many edges across it
- A (uniform) random walk on on the graph quickly converges to the stationary distribution

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph
- The graph satisfies a strong isoperimetric inequality
- Every set of vertices has many neighbours
- Every cut has many edges across it
- A (uniform) random walk on on the graph quickly converges to the stationary distribution
- The graph has a large spectral gap

The morality of expansion

In different contexts, an expander graph (or more accurately, an expander graph family) is defined morally in one of the following ways:

- The graph is a sparse approximation for the complete graph
- The graph satisfies a strong isoperimetric inequality
- Every set of vertices has many neighbours
- Every cut has many edges across it
- A (uniform) random walk on on the graph quickly converges to the stationary distribution
- The graph has a large spectral gap

Loosely speaking, under the right assumptions on the graphs, all of the above notions are equivalent.

Definition of Expander Graphs

We adopt the spectral point of view, restricted to regular graphs:

Definition ((n, d, λ)-graph)

We say that G is an (n, d, λ)-graph if G is a d-regular graph on n vertices such that

$$\lambda \geq \max_{i \neq 1} \frac{|\lambda_i|}{d}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ is the spectrum of the adjacency matrix A of G.

Definition of Expander Graphs

We adopt the spectral point of view, restricted to regular graphs:

Definition ((n, d, λ)-graph)

We say that G is an (n, d, λ)-graph if G is a d-regular graph on n vertices such that

$$\lambda \geq \max_{i \neq 1} \frac{|\lambda_i|}{d}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ is the spectrum of the adjacency matrix A of G.

Note here that $\lambda_1 = d$, and that the maximum on the right is 1 if and only if G is either disconnected or bipartite.

Definition (Expander Graphs)

A family of d-regular graphs $\{G_j\}_j$ is called an expander family with spectral gap $1 - \lambda$ (or a λ-expander family), if every G_j is an (n, d, λ)-graph for some $n = n_j$.

The Punchline: Expanders have Logarithmic Diameter

The usefulness of expanders in solving `ustcon` comes from the following lemma about their diameters:

Lemma (Diameter of an Expander)

Let $G = (V, E)$ be a connected (n, d, λ)-graph. Then,

$$\mathrm{diam}(G) = \max_{u,v} d(u, v) = \mathcal{O}_\lambda(\log n) = \mathcal{O}\left(\frac{\log n}{1-\lambda}\right)$$

where the implicit constant is effective, and efficiently computable.

Solving `ustcon` for λ-expanders

We will now describe a log-space algorithm for d-regular graphs for which every connected component is a λ-expander with $\lambda < 1$.

- Let (G, s, t) be the input.

Solving ustcon for λ-expanders

We will now describe a log-space algorithm for d-regular graphs for which every connected component is a λ-expander with $\lambda < 1$.

- Let (G, s, t) be the input.
- Let $\Delta = C_\lambda \log n$, where C_λ is the implicit constant from the lemma. Since every component is a λ-expander, if $s \rightsquigarrow t$, then there is a path of length of $\leq \Delta$.

Solving `ustcon` for λ-expanders

We will now describe a log-space algorithm for d-regular graphs for which every connected component is a λ-expander with $\lambda < 1$.

- Let (G, s, t) be the input.
- Let $\Delta = C_\lambda \log n$, where C_λ is the implicit constant from the lemma. Since every component is a λ-expander, if $s \rightsquigarrow t$, then there is a path of length of $\leq \Delta$.
- Each path $s = v_0, \cdots, v_k$ of length $k \leq \Delta$ can be encoded as a string in $\{1, \cdots, d\}^{\leq \Delta}$ where the jth letter encodes which of the d vertices adjacent to v_{j-1} is the next step, v_j.

Solving ustcon for λ-expanders

We will now describe a log-space algorithm for d-regular graphs for which every connected component is a λ-expander with $\lambda < 1$.

- Let (G, s, t) be the input.
- Let $\Delta = C_\lambda \log n$, where C_λ is the implicit constant from the lemma. Since every component is a λ-expander, if $s \rightsquigarrow t$, then there is a path of length of $\leq \Delta$.
- Each path $s = v_0, \cdots, v_k$ of length $k \leq \Delta$ can be encoded as a string in $\{1, \cdots, d\}^{\leq \Delta}$ where the jth letter encodes which of the d vertices adjacent to v_{j-1} is the next step, v_j.
- Keep a counter storing which of these paths we are on, initialized to all 1s. This takes $\mathcal{O}(\Delta \log d) = \mathcal{O}_\lambda(\log d \log n) = \mathcal{O}_{d,\lambda}(\log n)$ space.

Solving `ustcon` for λ-expanders

We will now describe a log-space algorithm for d-regular graphs for which every connected component is a λ-expander with $\lambda < 1$.

- Let (G, s, t) be the input.
- Let $\Delta = C_\lambda \log n$, where C_λ is the implicit constant from the lemma. Since every component is a λ-expander, if $s \rightsquigarrow t$, then there is a path of length of $\leq \Delta$.
- Each path $s = v_0, \cdots, v_k$ of length $k \leq \Delta$ can be encoded as a string in $\{1, \cdots, d\}^{\leq \Delta}$ where the jth letter encodes which of the d vertices adjacent to v_{j-1} is the next step, v_j.
- Keep a counter storing which of these paths we are on, initialized to all 1s. This takes $\mathcal{O}(\Delta \log d) = \mathcal{O}_\lambda(\log d \log n) = \mathcal{O}_{d,\lambda}(\log n)$ space.
- Trawl through this path to see if t appears. If it does, halt and accept. This takes $\mathcal{O}(1)$ space.

Solving ustcon for λ-expanders

We will now describe a log-space algorithm for d-regular graphs for which every connected component is a λ-expander with $\lambda < 1$.

- Let (G, s, t) be the input.
- Let $\Delta = C_\lambda \log n$, where C_λ is the implicit constant from the lemma. Since every component is a λ-expander, if $s \rightsquigarrow t$, then there is a path of length of $\leq \Delta$.
- Each path $s = v_0, \cdots, v_k$ of length $k \leq \Delta$ can be encoded as a string in $\{1, \cdots, d\}^{\leq \Delta}$ where the jth letter encodes which of the d vertices adjacent to v_{j-1} is the next step, v_j.
- Keep a counter storing which of these paths we are on, initialized to all 1s. This takes $\mathcal{O}(\Delta \log d) = \mathcal{O}_\lambda(\log d \log n) = \mathcal{O}_{d,\lambda}(\log n)$ space.
- Trawl through this path to see if t appears. If it does, halt and accept. This takes $\mathcal{O}(1)$ space.
- Increment the counter by 1, treating it as a d-ary integer. If the counter overflows to all 1s, then halt and reject.

Proof of Logarithmic Diameter

Proof.

Let $M = A/d$ be the random walk matrix, and $\vec{u} = \vec{v}_1 = (1/n, \cdots, 1/n)^T$ be the uniform distribution on the vertices. Let $\vec{p}$ be any probability distribution on the vertices, and let t be a time parameter.

Proof of Logarithmic Diameter

Proof.

Let $M = A/d$ be the random walk matrix, and $\vec{u} = \vec{v}_1 = (1/n, \cdots, 1/n)^T$ be the uniform distribution on the vertices. Let $\vec{p}$ be any probability distribution on the vertices, and let t be a time parameter.

$\vec{u}$ is an eigenvector of M with eigenvalue 1. Let $\vec{v}_2, \cdots \vec{v}_n$ be the eigenvector of A with eigenvalue $A\vec{v}_j = \lambda_j \vec{v}_j$.

Proof of Logarithmic Diameter

Proof.

Let $M = A/d$ be the random walk matrix, and $\vec{u} = \vec{v}_1 = (1/n, \cdots, 1/n)^T$ be the uniform distribution on the vertices. Let $\vec{p}$ be any probability distribution on the vertices, and let t be a time parameter.
$\vec{u}$ is an eigenvector of M with eigenvalue 1. Let $\vec{v}_2, \cdots \vec{v}_n$ be the eigenvector of A with eigenvalue $A\vec{v}_j = \lambda_j \vec{v}_j$. Then, clearly, for $j \neq 1$,

$$\|M\vec{v}_j\|_2 = \frac{|\lambda_j|}{d} \|\vec{v}_j\|_2 \leq \lambda \|\vec{v}_j\|_2$$

Proof of Logarithmic Diameter

Proof.

Let $M = A/d$ be the random walk matrix, and $\vec{u} = \vec{v}_1 = (1/n, \cdots, 1/n)^T$ be the uniform distribution on the vertices. Let $\vec{p}$ be any probability distribution on the vertices, and let t be a time parameter.

$\vec{u}$ is an eigenvector of M with eigenvalue 1. Let $\vec{v}_2, \cdots \vec{v}_n$ be the eigenvector of A with eigenvalue $A\vec{v}_j = \lambda_j \vec{v}_j$. Then, clearly, for $j \neq 1$,

$$\|M\vec{v}_j\|_2 = \frac{|\lambda_j|}{d} \|\vec{v}_j\|_2 \leq \lambda \|\vec{v}_j\|_2$$

In particular, this means that M is a λ-contraction on $\vec{u}^\perp$.

Proof of Logarithmic Diameter

Proof.

Let $M = A/d$ be the random walk matrix, and $\vec{u} = \vec{v}_1 = (1/n, \cdots, 1/n)^T$ be the uniform distribution on the vertices. Let $\vec{p}$ be any probability distribution on the vertices, and let t be a time parameter.

$\vec{u}$ is an eigenvector of M with eigenvalue 1. Let $\vec{v}_2, \cdots \vec{v}_n$ be the eigenvector of A with eigenvalue $A\vec{v}_j = \lambda_j \vec{v}_j$. Then, clearly, for $j \neq 1$,

$$\|M\vec{v}_j\|_2 = \frac{|\lambda_j|}{d} \|\vec{v}_j\|_2 \leq \lambda \|\vec{v}_j\|_2$$

In particular, this means that M is a λ-contraction on $\vec{u}^\perp$. Note that $(\vec{p} - \vec{u}) \cdot \vec{u} = 0$, and hence if $t \geq 1000(\frac{\log n}{\log(1/\lambda)}) \sim \frac{\log n}{1-\lambda}$,

$$\left\|M^t\vec{p} - \vec{u}\right\|_2 = \left\|M^t(\vec{p} - \vec{u})\right\|_2 \leq \lambda^t \|\vec{p} - \vec{u}\|_2 \leq 2\lambda^t \leq \frac{2}{n^{1000}}$$

Proof of Logarithmic Diameter

Proof.

Let $M = A/d$ be the random walk matrix, and $\vec{u} = \vec{v}_1 = (1/n, \cdots, 1/n)^T$ be the uniform distribution on the vertices. Let $\vec{p}$ be any probability distribution on the vertices, and let t be a time parameter.

$\vec{u}$ is an eigenvector of M with eigenvalue 1. Let $\vec{v}_2, \cdots \vec{v}_n$ be the eigenvector of A with eigenvalue $A\vec{v}_j = \lambda_j \vec{v}_j$. Then, clearly, for $j \neq 1$,

$$\|M\vec{v}_j\|_2 = \frac{|\lambda_j|}{d} \|\vec{v}_j\|_2 \leq \lambda \|\vec{v}_j\|_2$$

In particular, this means that M is a λ-contraction on $\vec{u}^\perp$. Note that $(\vec{p} - \vec{u}) \cdot \vec{u} = 0$, and hence if $t \geq 1000(\frac{\log n}{\log(1/\lambda)}) \sim \frac{\log n}{1-\lambda}$,

$$\left\|M^t \vec{p} - \vec{u}\right\|_2 = \left\|M^t(\vec{p} - \vec{u})\right\|_2 \leq \lambda^t \|\vec{p} - \vec{u}\|_2 \leq 2\lambda^t \leq \frac{2}{n^{1000}}$$

But this can only happen if every vertex has positive probability at time t. Thus, every vertex is reachable in t steps; $\operatorname{diam}(G) \leq t$ as desired.

Naive Idea: Graph Exponentiation

Definition (Graph Exponentiation)

For a graph G, we define the graph G^k as follows:

- $V(G^k) = V(G)$.

Naive Idea: Graph Exponentiation

Definition (Graph Exponentiation)

For a graph G, we define the graph G^k as follows:

- $V(G^k) = V(G)$.
- $A(G^k) = A(G)^k$.

Naive Idea: Graph Exponentiation

Definition (Graph Exponentiation)

For a graph G, we define the graph G^k as follows:

- $V(G^k) = V(G)$.
- $A(G^k) = A(G)^k$.

In other words, there is an edge between two vertices in G^k for every walk of length k between those two vertices in G.

Naive Idea: Graph Exponentiation

Definition (Graph Exponentiation)

For a graph G, we define the graph G^k as follows:

- $V(G^k) = V(G)$.
- $A(G^k) = A(G)^k$.

In other words, there is an edge between two vertices in G^k for every walk of length k between those two vertices in G.

Note s is connected to t in G^k if and only if it is connected in G, and it is immediate that if G is an (n, d, λ)-graph then G^k is an (n, d^k, λ^k)-graph. This vastly improves expansion – however, the degree blows up very quickly. Our algorithm (for constant degree expanders) does not apply!

More Refined Attempt: Tensoring Graphs

Definition (Tensor Product)

For graphs G and H, we have the tensor product $G \otimes H$:

- $V(G \otimes H) = V(G) \times V(H)$.

More Refined Attempt: Tensoring Graphs

Definition (Tensor Product)

For graphs G and H, we have the tensor product $G \otimes H$:

- $V(G \otimes H) = V(G) \times V(H)$.
- $A(G \otimes H) = A(G) \otimes A(H)$.

More Refined Attempt: Tensoring Graphs

Definition (Tensor Product)

For graphs G and H, we have the tensor product $G \otimes H$:

- $V(G \otimes H) = V(G) \times V(H)$.
- $A(G \otimes H) = A(G) \otimes A(H)$.

In other words, $uu' \sim vv'$ in $G \otimes H$ if $u \sim v$ in G and $u' \sim v'$ in H.

More Refined Attempt: Tensoring Graphs

Definition (Tensor Product)

For graphs G and H, we have the tensor product $G \otimes H$:

- $V(G \otimes H) = V(G) \times V(H)$.
- $A(G \otimes H) = A(G) \otimes A(H)$.

In other words, $uu' \sim vv'$ in $G \otimes H$ if $u \sim v$ in G and $u' \sim v'$ in H.

It is not hard to see that if G is an (n, d, α)-graph and H is an (m, d', β)-graph, then $G \otimes H$ is an (mn, dd', λ)-graph with $\lambda = \max\{\alpha, \beta\}$ – this follows from the fact that the spectrum of a tensor product of matrices is the pointwise product of their spectra.

The Replacement Product

This brings us to the novel construction known as the Zig-Zag product. For this, we assume that all graphs are on the vertex set $\{1, \cdots, n\}$ for some n.

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H, and for every $v \in V(G)$ fix an ordering on the edges incident on it. We define $G \textcircled{r} H$ as follows:

- $V(G \textcircled{r} H) = V(G) \times V(H)$

The Replacement Product

This brings us to the novel construction known as the Zig-Zag product. For this, we assume that all graphs are on the vertex set $\{1, \cdots, n\}$ for some n.

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H, and for every $v \in V(G)$ fix an ordering on the edges incident on it. We define $G \textcircled{r} H$ as follows:

- $V(G \textcircled{r} H) = V(G) \times V(H)$
- There is an edge between (u, i) and (u, j) in $G \textcircled{r} H$ for every edge between i and j in H. These are called zigs.

The Replacement Product

This brings us to the novel construction known as the Zig-Zag product. For this, we assume that all graphs are on the vertex set $\{1, \cdots, n\}$ for some n.

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H, and for every $v \in V(G)$ fix an ordering on the edges incident on it. We define $G \circledR H$ as follows:

- $V(G \circledR H) = V(G) \times V(H)$
- There is an edge between (u, i) and (u, j) in $G \circledR H$ for every edge between i and j in H. These are called zigs.
- There is an edge between (u, i) and (v, j) if the ith edge incident on u and the jth edge incident on v are the same edge. These are called zags.

The Replacement Product

This brings us to the novel construction known as the Zig-Zag product. For this, we assume that all graphs are on the vertex set $\{1, \cdots, n\}$ for some n.

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H, and for every $v \in V(G)$ fix an ordering on the edges incident on it. We define $G \circledR H$ as follows:

- $V(G \circledR H) = V(G) \times V(H)$
- There is an edge between (u, i) and (u, j) in $G \circledR H$ for every edge between i and j in H. These are called zigs.
- There is an edge between (u, i) and (v, j) if the ith edge incident on u and the jth edge incident on v are the same edge. These are called zags.
- All edges are either zigs or zags.

The Zig-Zag Product

We can now define the Zig-Zag product in terms of the Replacement product:

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H. We define $G \textcircled{r} H$ as follows:

- $V(G \textcircled{z} H) = V(G \textcircled{r} H) = V(G) \times V(H)$

The Zig-Zag Product

We can now define the Zig-Zag product in terms of the Replacement product:

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H. We define $G \circledR H$ as follows:

- $V(G \circledz H) = V(G \circledR H) = V(G) \times V(H)$
- There is an edge between (u, i) and (v, j) in $G \circledz H$ if there is a path of length three betwee the two $G \circledR H$ which is a zig-zag-zig.

The Zig-Zag Product

We can now define the Zig-Zag product in terms of the Replacement product:

Definition (Replacement Product)

For an (n, m, α)-graph G and an (m, d, β)-graph H. We define $G \circledr H$ as follows:

- $V(G \circledz H) = V(G \circledr H) = V(G) \times V(H)$
- There is an edge between (u, i) and (v, j) in $G \circledz H$ if there is a path of length three betwee the two $G \circledr H$ which is a zig-zag-zig.
- In other words, there is an edge between (u, i) and (v, j) in $G \circledz H$, whenever i is adjacent to k in H and j is adjacent to ℓ in H, where the kth edge at u is the same as the ℓth edge at v.

The Zig-Zag Product

We have the following theorem relating the expansion properties of $G \,ⓩ\, H$ with those of G and H:

Theorem (Reingold-Vadhan-Wigderson [RVW02])

Let G be an (n, m, α)-graph and H an (m, d, β)-graph. Then $G \,ⓩ\, H$ is an (nm, d^2, φ)-graph where $\varphi = \varphi(\alpha, \beta)$ satisfies the following:

The Zig-Zag Product

We have the following theorem relating the expansion properties of $G \textcircled{z} H$ with those of G and H:

Theorem (Reingold-Vadhan-Wigderson [RVW02])

Let G be an (n, m, α)-graph and H an (m, d, β)-graph. Then $G \textcircled{z} H$ is an (nm, d^2, φ)-graph where $\varphi = \varphi(\alpha, \beta)$ satisfies the following:

- $\alpha, \beta < 1$ *implies* $\varphi < 1$.

The Zig-Zag Product

We have the following theorem relating the expansion properties of $G \circledz H$ with those of G and H:

Theorem (Reingold-Vadhan-Wigderson [RVW02])

Let G be an (n, m, α)-graph and H an (m, d, β)-graph. Then $G \circledz H$ is an (nm, d^2, φ)-graph where $\varphi = \varphi(\alpha, \beta)$ satisfies the following:

- $\alpha, \beta < 1$ *implies* $\varphi < 1$.
- $\varphi(\alpha, \beta) \leq \alpha + \beta$

The Zig-Zag Product

We have the following theorem relating the expansion properties of $G \textcircled{z} H$ with those of G and H:

Theorem (Reingold-Vadhan-Wigderson [RVW02])

Let G be an (n, m, α)-graph and H an (m, d, β)-graph. Then $G \textcircled{z} H$ is an (nm, d^2, φ)-graph where $\varphi = \varphi(\alpha, \beta)$ satisfies the following:

- $\alpha, \beta < 1$ *implies* $\varphi < 1$.
- $\varphi(\alpha, \beta) \leq \alpha + \beta$
- $\varphi(\alpha, \beta) \leq 1 - \frac{(1-\beta^2)(1-\alpha)}{2}$

The Zig-Zag Product

We have the following theorem relating the expansion properties of $G \text{ⓩ} H$ with those of G and H:

Theorem (Reingold-Vadhan-Wigderson [RVW02])

Let G be an (n, m, α)-graph and H an (m, d, β)-graph. Then $G \text{ⓩ} H$ is an (nm, d^2, φ)-graph where $\varphi = \varphi(\alpha, \beta)$ satisfies the following:

- $\alpha, \beta < 1$ *implies* $\varphi < 1$.
- $\varphi(\alpha, \beta) \leq \alpha + \beta$
- $\varphi(\alpha, \beta) \leq 1 - \frac{(1-\beta^2)(1-\alpha)}{2}$

For now, note that if $\beta \leq 1/2$, then $(1 - \phi) \geq \frac{3}{8}(1 - \alpha)$.

Sketch of the Algorithm

We can now describe the algorithm. Suppose G is a d^{16}-regular graph (d will be fixed later) none of whose connected components are bipartite. Let H be a fixed (d^{16}, d, β)-graph with $\beta \leq 1/2$.

Sketch of the Algorithm

We can now describe the algorithm. Suppose G is a d^{16}-regular graph (d will be fixed later) none of whose connected components are bipartite. Let H be a fixed (d^{16}, d, β)-graph with $\beta \leq 1/2$.

- Recursively define $G_0 = G$ and $G_j = (G_{j-1} ⓩ H)^8$.

Sketch of the Algorithm

We can now describe the algorithm. Suppose G is a d^{16}-regular graph (d will be fixed later) none of whose connected components are bipartite. Let H be a fixed (d^{16}, d, β)-graph with $\beta \leq 1/2$.

- Recursively define $G_0 = G$ and $G_j = (G_{j-1} ⓩ H)^8$.
- It can be shown from our previous calculations that if all connected components of G_j are α_j-expanders for $\alpha_j < 1$, then

$$1 - \alpha_j \geq \min\left\{\frac{1}{2}, 2^j(1 - \alpha_0)\right\}$$

Sketch of the Algorithm

We can now describe the algorithm. Suppose G is a d^{16}-regular graph (d will be fixed later) none of whose connected components are bipartite. Let H be a fixed (d^{16}, d, β)-graph with $\beta \leq 1/2$.

- Recursively define $G_0 = G$ and $G_j = (G_{j-1} ⓩ H)^8$.
- It can be shown from our previous calculations that if all connected components of G_j are α_j-expanders for $\alpha_j < 1$, then

$$1 - \alpha_j \geq \min\left\{\frac{1}{2}, 2^j(1 - \alpha_0)\right\}$$

- It can be shown that each connected component of G has spectral gap at least $1/(d^{16}n^2)$. In particular, this means that for, say $j \geq 5\log(d^{16}n^2) = \mathcal{O}_d(\log n)$, $\alpha_j \leq 1/2$.

Sketch of the Algorithm

We can now describe the algorithm. Suppose G is a d^{16}-regular graph (d will be fixed later) none of whose connected components are bipartite. Let H be a fixed (d^{16}, d, β)-graph with $\beta \leq 1/2$.

- Recursively define $G_0 = G$ and $G_j = (G_{j-1} ⓩ H)^8$.
- It can be shown from our previous calculations that if all connected components of G_j are α_j-expanders for $\alpha_j < 1$, then

$$1 - \alpha_j \geq \min\left\{\frac{1}{2}, 2^j(1-\alpha_0)\right\}$$

- It can be shown that each connected component of G has spectral gap at least $1/(d^{16}n^2)$. In particular, this means that for, say $j \geq 5\log(d^{16}n^2) = \mathcal{O}_d(\log n)$, $\alpha_j \leq 1/2$.
- s, t is connected in G_0 only if s_j and t_j are connected in G_j where s_j and t_j are recursively chosen as any vertex in the cloud representing s_{j-1} and t_{j-1}. Check s_j, t_j connectivity instead.

Dotting i's

We need to check that everything actually can be done in log-space.

Dotting i's

We need to check that everything actually can be done in log-space.

- First note that if n_j is the size of the vertex set of G_j, then $n_j = nd^{16j}$. Thus, for $j \approx_d \log n$, $\log n_j \approx_d \log n$. Thus, connectivity in the iterated graph G_j is still log-space.

Dotting i's

We need to check that everything actually can be done in log-space.

- First note that if n_j is the size of the vertex set of G_j, then $n_j = nd^{16j}$. Thus, for $j \approx_d \log n$, $\log n_j \approx_d \log n$. Thus, connectivity in the iterated graph G_j is still log-space.
- We have the following lemma:

Lemma

Given two vertices u and v in G_j, it can be checked in $\mathcal{O}_d(\log n + j)$ space whether u and v are adjacent in G_j without explicitly storing the graph G_j.

Dotting i's

We need to check that everything actually can be done in log-space.

- First note that if n_j is the size of the vertex set of G_j, then $n_j = nd^{16j}$. Thus, for $j \approx_d \log n$, $\log n_j \approx_d \log n$. Thus, connectivity in the iterated graph G_j is still log-space.
- We have the following lemma:

Lemma

Given two vertices u and v in G_j, it can be checked in $\mathcal{O}_d(\log n + j)$ space whether u and v are adjacent in G_j without explicitly storing the graph G_j.

- Putting $j \approx_d \log n$ in the above lemma tells us that the algorithm for connectivity in G_j can be executed with log-space overhead without ever constructing G_j.

Dotting i's

We need to check that everything actually can be done in log-space.

- First note that if n_j is the size of the vertex set of G_j, then $n_j = nd^{16j}$. Thus, for $j \approx_d \log n$, $\log n_j \approx_d \log n$. Thus, connectivity in the iterated graph G_j is still log-space.
- We have the following lemma:

Lemma

Given two vertices u and v in G_j, it can be checked in $\mathcal{O}_d(\log n + j)$ space whether u and v are adjacent in G_j without explicitly storing the graph G_j.

- Putting $j \approx_d \log n$ in the above lemma tells us that the algorithm for connectivity in G_j can be executed with log-space overhead without ever constructing G_j.
- Finally, a graph $(d^{16}, d, 1/2)$ can be found using the probabilistic method, for some fixed small d.

Crossing t's

We have solved the problem for graphs G which which are d^{16}-regular, and none of whose components are bipartite. What about general graphs?

Crossing t's

We have solved the problem for graphs G which which are d^{16}-regular, and none of whose components are bipartite. What about general graphs?

- First, we reduce to the 3-regular case: for a general graph G, create G' by replacing a vertex v with a cycle of length of $\deg(v)$, and edges of G by matchings (that is, if the ith edge of u and the jth edge of v coincide then draw and edge between (u, i) and (v, j)).

Crossing t's

We have solved the problem for graphs G which which are d^{16}-regular, and none of whose components are bipartite. What about general graphs?

- First, we reduce to the 3-regular case: for a general graph G, create G' by replacing a vertex v with a cycle of length of $\deg(v)$, and edges of G by matchings (that is, if the ith edge of u and the jth edge of v coincide then draw and edge between (u, i) and (v, j)).
- Add enough loops at every point to make it a $d^{16} - 3$ regular graph. This does not change connectivity.

Crossing t's

We have solved the problem for graphs G which which are d^{16}-regular, and none of whose components are bipartite. What about general graphs?

- First, we reduce to the 3-regular case: for a general graph G, create G' by replacing a vertex v with a cycle of length of $\deg(v)$, and edges of G by matchings (that is, if the ith edge of u and the jth edge of v coincide then draw and edge between (u, i) and (v, j)).
- Add enough loops at every point to make it a $d^{16} - 3$ regular graph. This does not change connectivity.

We now return to the intuition behind Reingold-Vadhan-Wigderson.

Why is $G \textcircled{z} H$ a good expander?

There are two basic intuitions, both of which can be converted into (somewhat technical) proofs:

Why is $G \textcircled{z} H$ a good expander?

There are two basic intuitions, both of which can be converted into (somewhat technical) proofs:

1. H is a low-degree good expander, so it can be thought of as a good approximation to the regular graph. Thus, $H \approx K_m$. Then, $G \textcircled{z} H \approx G \otimes K_m$, and this tells us that $G \textcircled{z} H$ must have better expansion than G.

Why is $G \textcircled{z} H$ a good expander?

There are two basic intuitions, both of which can be converted into (somewhat technical) proofs:

1. H is a low-degree good expander, so it can be thought of as a good approximation to the regular graph. Thus, $H \approx K_m$. Then, $G \textcircled{z} H \approx G \otimes K_m$, and this tells us that $G \textcircled{z} H$ must have better expansion than G.
2. Alternatively, if a random walk on $G \textcircled{z} H$ has not converged to the stationary distribution, then we have two extremal cases:

Why is $G ⓩ H$ a good expander?

There are two basic intuitions, both of which can be converted into (somewhat technical) proofs:

1. H is a low-degree good expander, so it can be thought of as a good approximation to the regular graph. Thus, $H \approx K_m$. Then, $G ⓩ H \approx G \otimes K_m$, and this tells us that $G ⓩ H$ must have better expansion than G.
2. Alternatively, if a random walk on $G ⓩ H$ has not converged to the stationary distribution, then we have two extremal cases:
    - The probability is well distributed among clouds, but are badly distributed within each cloud. In this case, each step will use zigs, thus improving the distribution within each cloud, while the zag will only permute the total density on each cloud.

Why is $G \textcircled{z} H$ a good expander?

There are two basic intuitions, both of which can be converted into (somewhat technical) proofs:

1. H is a low-degree good expander, so it can be thought of as a good approximation to the regular graph. Thus, $H \approx K_m$. Then, $G \textcircled{z} H \approx G \otimes K_m$, and this tells us that $G \textcircled{z} H$ must have better expansion than G.
2. Alternatively, if a random walk on $G \textcircled{z} H$ has not converged to the stationary distribution, then we have two extremal cases:
    - The probability is well distributed among clouds, but are badly distributed within each cloud. In this case, each step will use zigs, thus improving the distribution within each cloud, while the zag will only permute the total density on each cloud.
    - Each cloud is well-distributed, but the probability is badly distributed among clouds. In this case, the zigs will not change the distribution within each cloud, while the zag will improve distribution among the clouds.

Why is $G \textcircled{z} H$ a good expander?

There are two basic intuitions, both of which can be converted into (somewhat technical) proofs:

1. H is a low-degree good expander, so it can be thought of as a good approximation to the regular graph. Thus, $H \approx K_m$. Then, $G \textcircled{z} H \approx G \otimes K_m$, and this tells us that $G \textcircled{z} H$ must have better expansion than G.
2. Alternatively, if a random walk on $G \textcircled{z} H$ has not converged to the stationary distribution, then we have two extremal cases:
    - The probability is well distributed among clouds, but are badly distributed within each cloud. In this case, each step will use zigs, thus improving the distribution within each cloud, while the zag will only permute the total density on each cloud.
    - Each cloud is well-distributed, but the probability is badly distributed among clouds. In this case, the zigs will not change the distribution within each cloud, while the zag will improve distribution among the clouds.
    - The general case is a superposition of the above two extremes.

Thank You!