A Theorist's Toolkit **(CMU 18-859T, Fall 2013)**

Lecture 12: Expanders

October 16, 2013

Lecturer: Ryan O'Donnell *Scribe: Aleksandr M. Kazachkov*

1 Overview

This lecture will talk about expanders. Expanders arise frequently in many areas of CS theory. See, for instance, the examples and references from [Alo86], or the more recent survey by Hoory et al. [HLW06]. Loosely, expanders are a family of graphs that are:

1. Sparse
2. Highly connected
3. Explicitly constructible

The third property is included because random graphs are likely to have the first two properties. We do not want a random graph; we want it to be deterministically and efficiently constructible.

There are many definitions used for expanders. We will always talk about an undirected graph $G = (V, E)$. Here, $|V| = n$, where we consider $n \to \infty$, and the graph is d-regular (i.e., $\deg(u) = d$ for all $u \in V$), where we think of d as a constant.

Note that n goes to infinity, but d is thought of as a constant, so the graph automatically becomes sparse as n grows large, since $|E| = \frac{dn}{2} = O(n)$.

What does "highly connected" mean? We could talk about this either in terms of edge expansion or vertex expansion. Informally, a graph is highly connected when every $S \subseteq V$: $0 < |S| \leq \frac{n}{2}$ has many edges (for edge expansion) or vertices (for vertex expansion) on its boundary. Instead of half the vertices, we sometimes define an expander as holding for all sets S of size at most some other fraction of n.

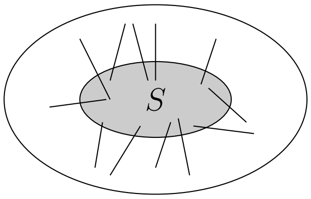

Figure 1: A picture demonstrating how we picture expansion. The lines coming from S represent edges from vertices in S to those in $\bar{S}$. We either count the number of edges leaving S or the number of neighbors of S.

For example, in the edge expansion case, for every such subset S of the vertices, we would like the number of edges going from S to $\bar{S}$, the complement of S, to be large: $|\partial S| = \left|E(S, \bar{S})\right| = \Omega(|S|)$. Note that we also have $\left|E(S, \bar{S})\right| \leq d\,|S| = O(|S|)$. For vertex expansion, we talk about the number of vertices from $\bar{S}$ that can be reached via one step from a vertex in S. These vertices are called the neighborhood of S, which will be denoted by $N(S)$. We have the same property that $|N(S)| \leq d\,|S|$, and we would like it also $\Omega(|S|)$.

For many of the examples in this lecture, we will talk about expanders in the context of left d-regular bipartite graphs, which are graphs $G = (V, E)$ where V is partitioned into two disjoint sets L and R. The d-regularity of the graph is required to be maintained on the left set of vertices, L, and the expansion is chosen from sets $S \subseteq L$. Random left d-regular bipartite graphs are simple to construct by choosing d neighbors from R for each vertex in L. If we wanted to have d-regularity on both sides, we could construct the graph by taking a union of d random perfect matchings, which is also a method that could be used to construct d-regular general graphs. (This could yield parallel edges, but if this is a concern, it can be shown that this happens with low probability.)

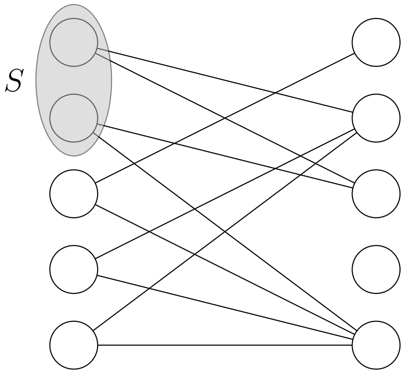

Figure 2: An example of a left d-regular graph, where $d = 2$ and $n = 5$. If we take the circled two nodes on the left as S, then the number of edges leaving it is 3.

More formally, we can use one of the following definitions for an expander family.

Definition 1.1. A family of expander graphs is a family of d-regular graphs, given a fixed positive integer d, such that if G_n is a member of the family with vertex set $V = [n]$, then G_n has good edge expansion for all $S \subseteq V : 0 < |S| \leq \frac{1}{2}n$:

$$\mathop{\mathbf{Pr}}_{u \sim v}[u \in S, v \notin S] \geq \epsilon,$$

where ϵ is some constant greater than zero. The restriction of S to be at most half the vertices can be changed to a different constant proportion of the vertices.

Definition 1.2. A family of expander graphs is a family of d-regular graphs, given a fixed positive integer d, such that if G_n is a member of the family with vertex set $V = [n]$, then G_n has good vertex expansion for all $S \subseteq V : 0 < |S| \leq \frac{1}{2}n$:

$$|N(S)| \geq \epsilon \cdot d \cdot |S|\,,$$

where ϵ is some constant greater than one. The restriction of S to be at most half the vertices can be changed to a different constant proportion of the vertices.

In Section 2, random constructions of expanders are shown. Afterward, in Section 3, we discuss applications of expanders. Finally, in Section 4, we discuss obtaining explicit constructions of expanders.

2 Random constructions

As mentioned in Section 1, if only sparsity and high connectivity were desired, these properties could be achieved by random graphs. Thus, the ideal we strive for is to construct, explicitly, families of graphs with the same nice expander properties as random graphs.

Example 2.1 ([Pin73])**.** *Consider, for all $d \geq 3$ and for all $n \geq n_0$ for some constant n_0, a random left d-regular $(n+n)$-vertex bipartite graph. This graph has, with high probability,*

$$|N(S)| \geq (d-2)\,|S|\,,$$

for all $|S| \leq c_d \cdot n$ and $S \subseteq L$. Here, $(d-2)$ can be replaced by $(d-a)$ for any $a > 1$, and c_d is a constant dependent on d (mentioned to be $\sim \frac{1}{20d^4}$).

The proof proceeds by a straightforward application of the probabilistic method and some approximation of binomial coefficients.

Example 2.2. *[Bas81] We again consider a random left d-regular bipartite graph, where $d \geq 64$. Let $|L| = n$ and $|R| = \frac{3}{4}n$. Then*

$$|N(S)| \geq .8d\,|S|\,, \text{ for all } S \subseteq L : |S| \leq \frac{.02}{d}n.$$

This is also proved by the probabilistic method.

Example 2.1 is due to Pinkser [Pin73]. The numbers .8 and .02 may vary depending on different proofs and applications. This was the first paper to prove that an expander family exists. An explicit construction is more difficult to find and was first provided by Margulis [Mar73].

Definition 2.3. An **explicit** construction of an expander provides, in deterministic poly(n) time, the entire adjacency matrix.

A **strongly explicit** construction of an expander is one that can provide in poly log(n) time, for any $u \in V = [n]$ and $i \in [d]$, the vertex v that is the ith neighbor of u. This is polynomial in the input u and i, which has size $\lg n + \lg d$.

Note that a strongly explicit construction does not give the full graph, but rather the specified neighbor of a given vertex in the graph. Thus, it allows us to work with graphs that have an exponential (in n) number of vertices.

The informal statement is that for "almost" all expander variants (bipartite or non-bipartite, edge or vertex expansion), we know explicit constructions which "almost" retain parameters obtained by random graphs.

3 Applications

3.1 Coding theory

The first application will be to obtaining asymptotically good codes. We show these can be obtained from good expanders, and they will have the properties:

- positive constant rate,
- positive constant minimum distance, and
- efficient to decode and encode.

Say we have an explicit algorithm for Example 2.2, and let $d = 64$. Thus, for all $S \subseteq L : |S| \leq \frac{.02}{64}n$, we have that $|N(S)| \geq .8d\,|S|$.

Claim 3.1. *If $0 \neq |S| \leq \frac{.02}{64}n$, there exists a $v \in N(S)$ with exactly one neighbor in S.*

Proof. If, for all $v \in N(S)$, $|N(v) \cap S| \geq 2$, then, since $|N(S)| \geq .8d\,|S|$,

$$|E(S, N(S))| \geq 2\,|N(S)| \geq 2 \cdot .8 \cdot 64 \cdot |S| > 64\,|S|\,.$$

This is a contradiction, since the left partition is 64-regular, so $|E(S, N(S))| \leq 64\,|S|$. □

Now we introduce the Tanner code [Tan84], constructed from the expander generated by the explicit algorithm for Example 2.2. Consider the code whose parity check matrix H (where the elements of the code are all length ℓ strings z such that $Hz = 0$) is the adjacency matrix of the expander graph we constructed. The entries of the adjacency matrix are from $\mathbb{F}_2^{|R| \times |L|}$.

The message length / dimension / rank of this code is $|L| - |R| = n - \frac{3}{4}n = \frac{1}{4}n$. Thus, this is a $[n, \frac{1}{4}n, D]_2$ code, for some minimum distance D to be determined. The rate is $\frac{1}{4}$, which is constant, as promised.

Claim 3.2. *The minimum distance of the code D is greater than $\frac{.02}{64}n$ (assume $\frac{.02}{64}$ is an integer). Note that this is also an absolute constant, as desired.*

Proof. Assume for the sake of contradiction that the minimum distance at most $\frac{.02}{64}n$. Thus, there exists a non-zero codeword z with Hamming weight $|z| \leq \frac{.02}{64}n$. I.e., z has at most that many non-zero components. Let $S = \{u \in [n] : z_u = 1\}$. Since z is non-zero, we know that $S \neq \emptyset$, and the Hamming weight of z implies $|S| \leq \frac{.02}{64}n$.

Since we are multiplying H by z, the non-zero entries of z correspond to the columns of H representing vertices in S. By Claim 3.1, we have that there exists a $v \in N(S)$ with exactly one neighbor in S. This corresponds to a row of H with exactly one non-zero entry in the columns indexed by S. But this implies that the vth entry of Hz is 1, which contradicts that z is a codeword ($Hz = 0$ for all codewords). □

The next proposition confirms that Tanner codes are decodable efficiently. It critically uses that the .8 in Example 2.2 is at least 3/4.

Proposition 3.3. *Tanner codes are decodable from up to at most $\frac{D}{2}$ errors in* poly(n) *time. It can be done in $O(n)$ time with a more careful algorithm.*

The algorithm used for the weaker runtime in Proposition 3.3 is simple. Let z not be a codeword. Thus, $Hz \neq 0$. We use a variant of what is called a *belief propagation* algorithm.

Algorithm 3.1 Algorithm for Proposition 3.3

```
Input: $z$ such that $Hz \neq 0$
Output: Word $z$ at most $D/2$ from being a codeword
  while $Hz \neq 0$ do
      Flip any $z_i$ that decreases the Hamming weight of $Hz$
  end while
  return $z$
```

This gives a vector at most $D/2$ from a codeword.

3.2 Deterministic error reduction

Last time, we studied derandomizing randomized algorithms while retaining the same properties of the algorithm. In this application, we seek *deterministic error reduction*: using not too many more random bits (continuing to view them as a resource, as in the previous lecture), we want to reduce the error of a randomized algorithm

Let $\mathcal{A}$ be a randomized one-sided algorithm for deciding membership in a language $\mathcal{P}$ that runs in time T. E.g., if the input is a number x, $\mathcal{P}$ could be "is x a prime number?" and $\mathcal{A}$ could be the Miller-Rabin primality test. If $x \in \mathcal{P}$, then $\mathbf{Pr}[\mathcal{A} \text{ says } \texttt{YES}] = 1$. If $x \notin \mathcal{P}$, then $\mathcal{A}$ is still correct with high probability, say, $\mathbf{Pr}[\mathcal{A} \text{ says } \texttt{YES}] \leq .01$.

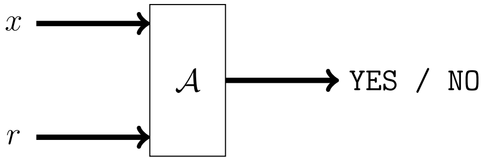

Figure 3: The inputs and outputs for $\mathcal{A}$ deciding membership in a language $\mathcal{P}$. Here x is the input for which we want to check membership in $\mathcal{P}$, r is an n-bit seed string from $\{0,1\}^n$, the algorithm runs in time T, and the output is either YES or NO.

What are ideas for reducing error?

1. Repeat the entire algorithm d times, with independent random input seed strings $r_1, \ldots, r_d$. This reduces the error but uses many random bits:
   (a) Uses $dT + O(n)$ time (where the $O(n)$ comes from regenerating the random string)
   (b) Uses dn random bits
   (c) Yields error probability at most $.01^d$

2. Say we have a strongly explicit algorithm for Example 2.2. Take this expander constructed from Example 2.2, only with $|L| = |R| = 2^n$ (both partitions now have equal size). The index of a vertex within one of the sides is then an n-bit string.

   The algorithm proceeds as follows. Pick a random vertex $\ell \in L$. This costs n random bits. Next, compute, using the strongly explicit algorithm, the d neighbors of ℓ: $r_1, \dots, r_d$ that form $N(S) \subseteq R$. This costs $\text{poly}\log(2^n) = \text{poly}(n)$ time. We now have d n-bit strings, with which we can run $\mathcal{A}$. We output `YES` if the answer is positive for all the trials, and otherwise we output `NO`. How does this do?

   (a) Uses time $dT + \text{poly}(n)$

   (b) Uses n random bits

   (c) Yields error at most $\frac{.02}{d}$

   Note that this error reduction is as good as $.01^d$, but the amazing thing is we do not use any more random bits!

We now do the analysis for the error that the expander method yields. For any x, $\mathcal{A}$ is only wrong on a .01 fraction of random seeds. Let $B_x \subseteq R$ denote this 1% of "bad" strings causing $\mathcal{A}$ to error.

Let $S \subseteq L$ be the "bad" choices for $\ell \in L$: those ℓ for which $N(\ell) \subseteq B_x$ (otherwise running our algorithm will give at least one `NO`, so it will not err).

Claim 3.4. $|S| < \frac{.02}{d} 2^n$.

Proof. Assume for the sake of contradiction that $|S| \geq \frac{.02}{d} 2^n$. Let $S' \subseteq S$ such that $|S'| = \frac{.02}{d} 2^n$. By the expander properties, for any subset T of L satisfying $|T| \leq \frac{.02}{d} 2^n$, we have $|N(T)| \geq .8d\,|T|$. Thus,

$$|N(S')| \geq .8d\,|S'| = .8 \cdot .02 \cdot 2^n > .012^n = |B_x|\,.$$

But then there exists a vertex $v \in N(S') \setminus B_x$, for which $\mathcal{A}$ would say `NO` and thus our error-reduction algorithm would also be correct. □

Thus, our error probability is $\leq \frac{.02}{d}$ (probability we choose a "bad" ℓ is at most this).

A third method for error-reduction is taking a t-step random walk from a random vertex. This uses $n + t \log d$ random bits and yields an error of at most $2^{-\theta(t)}$.

4 Explicit constructions

In this section, we discuss how to deterministically obtain expanders for which we have already given probabilistic constructions. The parameters for the expanders are whether the graph is (1) bipartite or non-bipartite, and (2) required to have edge versus vertex expansion. Our examples used bipartite graphs, but generally we talk about non-bipartite graphs.

As seen in Definition 1.1, edge expansion is when

$$\forall\ S \subseteq V : |S| \leq \frac{n}{2}, \quad \mathop{\mathbf{Pr}}_{u\sim v}[u \in S, v \notin S] \geq \epsilon.$$

Vertex expansion has a similar definition, as seen in Definition 1.2:

$$\forall\ S \subseteq V : |S| \leq \frac{n}{2}, \quad |N(S)| \geq \epsilon \cdot d \cdot |S|\,.$$

These properties can be hard to check, but luckily, we have already seen a relationship that can be used to make them computationally easier. Namely, $\mathbf{Pr}_{u\sim v}[u \in S, v \notin S] \leq 2\sqrt{\lambda_1}$, by Cheeger's inequality from Lecture 8, where λ_1 is the second-smallest eigenvalue of the normalized Laplacian matrix of G, L_G. There is a similar inequality due to Alon and Milman [AM84] for approximating vertex expansion by relating it to the second eigenvalue.

This leads to a third commonly-used definition of expansion, which requires that λ_1 is large. These correspond to graphs with no small cuts, and also those that mix well, as we have seen in Lecture 8.

Definition 4.1. An (n, d, ϵ)-spectral expander is an n-vertex, d-regular graph with $\lambda_1 \geq \epsilon$.

As mentioned before, the first explicit construction of expanders was given by Margulis [Mar73], who gave it for a continuous analog. Later, Gabber and Galil [GG81] provided a discrete version, and indeed a strongly explicit algorithm for constructing these expanders. We next describe this construction.

Let G be a graph with vertex set $V = \mathbb{Z}_m^2$ and edge set E in which we connect (x, y) to:

1. $(x \pm y, y)$,

2. $(x \pm (y + 1), y)$,

3. $(x, y \pm x)$,

4. $(x, y \pm (x + 1))$.

These are eight neighbors for each vertex, making the graph 8-regular, and it takes $O(\log n)$ time to compute the neighbors. Note the graph is undirected. If (a, b) has an edge to (c, d), then either $a = c$ or $b = d$. Suppose first that $b = d$. If $c = a + b$ or $c = a - b$, then one of $(c \pm d, d)$ yields (a, b), so the edge exists in the other direction as well. If instead $c = a + (b + 1)$, then $(c - (d + 1), d) = (a + (b + 1) - (b + 1), b) = (a, b)$; if $c = a - (b + 1)$, then $(c + (d + 1), d) = (a - (b + 1) + (b + 1), b) = (a, b)$. The symmetric argument holds for if $a = b$.

Theorem 4.2 ([GG81]). *The graph as described above is an $(m^2, 8, \epsilon)$-spectral expander for some $\epsilon > 0$ ($\approx .01$).*

In Homework 3, we saw that a K_n has all eigenvalues (other than λ_0) large. The next theorem shows that a random d-regular graph is likely to have a high λ_i, for all $i \in [n - 1]$.

Theorem 4.3 ([Fri03])**.** *In a random n-vertex d-regular graph G, all eigenvalues of L_G (other than $\lambda_0 = 0$) fall, with probability $\geq 1 - o_n(1)$, in the range*

$$\left[1 \pm \frac{2}{\sqrt{d}}\sqrt{1 - \frac{1}{d} + \epsilon}\right].$$

If we look at the eigenvalues of K, the normalized adjacency matrix of G, we have shown in Lecture 8 that $\kappa_i = 1 - \lambda_i$ so that $1 = \kappa_0 \geq \kappa_1 \geq \cdots \geq \kappa_{n-1} \geq -1$. Therefore, by Theorem 4.3,

$$|\kappa_i| \leq \kappa := \frac{2}{\sqrt{d}}\sqrt{1 - \frac{1}{d} + \epsilon} \xrightarrow[d \to \infty]{} 0. \tag{1}$$

Lemma 4.4 (Expander Mixing Lemma [AC88])**.** *For a graph that satisfies the property 1, we have that for all $S, T \subseteq V$,*

$$\left|\mathop{\mathbf{Pr}}_{u \sim v}[u \in S, v \in T] - \mathrm{vol}(S)\,\mathrm{vol}(T)\right| \leq \kappa\sqrt{\mathrm{vol}(S)\,\mathrm{vol}(T)}.$$

We can rewrite the left side of Lemma 4.4 as

$$\mathrm{vol}(S)\,\mathrm{vol}(T) = \mathop{\mathbf{Pr}}_{\substack{u \sim \pi \\ v \sim \pi}}[u \in S, v \in T],$$

to give the interpretation of the lemma as “choosing a random edge is almost the same as choosing endpoints independently.”

We next discuss *Ramanujan graphs.*

Definition 4.5. A Ramanujan graph is a d-regular graph that satisfies

$$\kappa := \max\{|\kappa_i| : i \in [n-1]\} \leq \frac{2}{\sqrt{d}}\sqrt{1 - \frac{1}{d}}.$$

Theorem 4.6 ([LPS88])**.** *There exists an explicit construction of Ramanujan graphs if d is one more than a prime power and $n = 1 + p$ for p prime, and $p \equiv 1 \pmod 4$.*

The construction in Theorem 4.6 is only somewhat explicit, since it depends on the given prime, but we do not know how to deterministically construct a prime.

The next corollary is folklore, in the sense that it follows from the paper that provided Theorem 4.6, but it is seemingly not explicitly in this paper.

Corollary 4.7. *Let $V = \mathbb{Z}_p$, and E be the set of edges in which we connect $a \in V$ to $a + 1$, $a - 1$, and $\frac{1}{a}$ modulus p, where we define $\frac{1}{0}$ to be 0. Then this is a $(p, 3, \epsilon)$-spectral expander (where $\epsilon \approx .01$).*

Once you have p, the construction of the expander in Corollary 4.7 is strongly explicit.

Next, we give an example of a construction related to the last question of Homework 3. The original paper [RVW00] used the zig-zag product of two graphs, but we can equivalently use the simpler replacement product from the homework.

Definition 4.8. Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, where G is an n-vertex, D-regular graph, and H is a D-vertex, d-regular graph, the replacement product $G \textcircled{r} H$ is defined as the $2d$-regular graph with vertex set $V_G \times V_H$, in which each vertex of G is replaced by a copy of H, and (g, h) having an edge to (g', h') if and only if either (1) $g = g'$ and $(h, h') \in E_H$, or (2) $g \neq g'$ and g' is the hth neighbor of g in G, and g is the h'th neighbor of g' in G. For case (2), we add d parallel edges.

Consider for example $G = K_3$ and $H = K_2$. Thus, $n = 3$, $D = 2$, and $d = 1$. Then $G \textcircled{r} H$ is a 6-cycle:

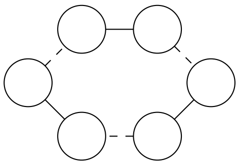

Figure 4: This is the replacement product of a K_3 and K_2. The dashed lines indicate edges from a vertex of one copy of the K_2 to another copy of the K_2, while the solid lines are edges within the same copy of the K_2.

Let G be an (n, D, ϵ_G)-spectral expander, and let H be a (D, d, ϵ_H)-spectral expander.

Theorem 4.9 ([JSTV04]). *The replacement product $G \textcircled{r} H$ is a $(Dn, 2d, \frac{\epsilon_H \epsilon_G}{16})$-spectral expander.*

This is great because we can make the number of vertices go up, the degree go down, but the spectral expansion also goes down. So we need to boost it back up.

Given G, look at G^t (the graph after a t-step random walk: connect $u \sim v$ if there exists a t-step path from u to v). So the degree in G^t is d^t. This gets us a $(Dn, (2d)^t, \approx t\frac{\epsilon_H \epsilon_G}{16})$-spectral expander.

However then the degree goes up, so we can iterate the operator $G^t \textcircled{r} H$. This paradigm can be used to prove the PCP Theorem, it turns out.

References

[AC88] N. Alon and F. R. K. Chung. Explicit construction of linear sized tolerant networks. In *Proceedings of the First Japan Conference on Graph Theory and Applications (Hakone, 1986)*, volume 72, pages 15–19, 1988.

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. Theory of computing (Singer Island, FL, 1984).

[AM84] N. Alon and V. D. Milman. Eigenvalues, expanders and superconcentrators. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science, 1984*, SFCS '84, pages 320–322, Washington, DC, USA, 1984. IEEE Computer Society.

[Bas81] L. A. Bassalygo. Asymptotically optimal switching circuits. *Problems of Information Transmission*, 17(3):206–211, 1981.

[Fri03] J. Friedman. A proof of Alon's second eigenvalue conjecture. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 720–724 (electronic), New York, 2003. ACM.

[GG81] O. Gabber and Z. Galil. Explicit constructions of linear-sized superconcentrators. *J. Comput. System Sci.*, 22(3):407–420, 1981. Special issued dedicated to Michael Machtey.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561 (electronic), 2006.

[JSTV04] M. Jerrum, J. Son, P. Tetali, and E. Vigoda. Elementary bounds on Poincaré and log-Sobolev constants for decomposable Markov chains. *Ann. Appl. Probab.*, 14(4):1741–1765, 2004.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[Mar73] G. A. Margulis. Explicit constructions of expanders. *Problemy Peredači Informacii*, 9(4):71–80, 1973.

[Pin73] M. S. Pinsker. On the complexity of a concentrator. *7th International Teletraffic Conference*, pages 318/1–318/4, 1973.

[RVW00] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings of 41st Annual Symposium on Foundations of Computer Science*, pages 3–13, 2000.

[Tan84] M. R. Tanner. Explicit concentrators from generalized N-gons. *SIAM J. Algebraic Discrete Methods*, 5(3):287–293, 1984.