Expander Graphs and their Applications*
Lecture 3: Random Walks on Expander Graphs

Notes taken by Boaz Barak and Udi Wieder

March 2002

Summary: In this lecture we consider random walks on expander graphs. We will see that the t vertices on a length t random walk on a expander graph "look like" (in some respects) t random independently chosen vertices. This occurs even though sampling a length t walk on a (constant-degree) expander requires a significantly smaller number of random bits than sampling t random vertices. We will use these properties for two applications. The first application is a randomness-efficient error reduction procedure for randomized algorithms. The second application is proving a strong hardness-of-approximation result for the maximum clique problem.

1 Preliminaries

(n, d, α) graphs. For a graph G on n vertices we denote by $\lambda_0(G), \ldots, \lambda_{n-1}(G)$ the eigenvalues of the adjacency matrix of G, where $\lambda_0(G) \geq \lambda_1(G) \geq \ldots \geq \lambda_{n-1}(G)$ (recall that all the eigenvalues are real numbers since G is undirected and so the adjacency matrix is symmetric). We say that a graph G on n vertices is an *(n, d)-graph* if it is d-regular. In this case $\lambda_0(G) = d$. For a number $\alpha < 1$, we say that G is an *(n, d, α)-graph* if G is an (n, d)-graph and $\max(|\lambda_1(G)|, |\lambda_{n-1}(G)|) \leq \alpha d$.

Vectors and norms. For two vectors $\vec{u}, \vec{v} \in \mathbb{R}^n$, we define the *dot product* of $\vec{u}$ and $\vec{v}$, denoted $\langle \vec{u}, \vec{v} \rangle$ to be $\sum_{i=1}^{n} \vec{u}_i \vec{v}_i$. For a vector $\vec{u} \in \mathbb{R}^n$ we define the l_1, l_2 and l_∞ norms of $\vec{u}$ as:

$$\|\vec{u}\|_1 \stackrel{def}{=} \sum_{i=1}^{n} |\vec{u}_i|$$

$$\|\vec{u}\|_2 \stackrel{def}{=} \sqrt{\langle \vec{u}, \vec{u} \rangle} = \left(\sum_{i=1}^{n} \vec{u}_i^2 \right)^{1/2}$$

$$\|\vec{u}\|_\infty \stackrel{def}{=} \max_{1 \leq i \leq n} |\vec{u}_i|$$

Probability vectors. We say that a vector $\vec{p} \in \mathbb{R}^n$ is a *probability vector* if for every $1 \leq i \leq n$, $\vec{p}_i \geq 0$ and $\sum_{i=1}^{n} \vec{p}_i = 1$. We denote by $\vec{u}$ the probability vector that corresponds to the uniform distribution. That is, $\vec{u} = \frac{1}{n}(1, \ldots, 1)$.

* Lecture Notes for a course given by Nati Linial and Avi Wigderson at the Hebrew University, Israel.

2 A random walk on an expander is rapidly mixing.

In this section we show that a random walk on the vertices of an expander mixes rapidly towards the stationary distribution. Let G be an (n, d, α) expander, and let A be it's adjacency matrix.

Definition 1. A random walk over the vertices of G is a stochastic process defining a series of vertices $(X_0, X_1, \dots)$ in which X_0 is a vertex of G chosen by some initial distribution and X_{i+1} is chosen uniformly at random from the neighbors of X_i.

A random walk is in fact a Markov chain where the set of states of the chain is the set of vertices of the graph.

Definition 2. The *normalized adjacency matrix* of G is defined to be $\frac{1}{d}A$ and is denoted by $\hat{A}$.

The following facts are easy to verify:

1. $\hat{A}$ is double stochastic; i.e. every column and every row sums up to 1.
2. Denote by $\hat{\lambda}_0, \dots, \hat{\lambda}_n$ the eigenvalues of $\hat{A}$, then $\hat{\lambda}_0 = 1$ and $\max\{|\hat{\lambda}_1|, |\hat{\lambda}_n|\} = \alpha$.

$\hat{A}$ can be viewed as the transition matrix of the Markov chain defined by the random walk over the vertices of G. In other words let X be a random vertex in G with probability vector $\vec{p}$. Let Y be a uniformly chosen neighbor of X. We claim that the probability vector of Y is given by $\hat{A}\vec{p}$. To see this write the Bayesian equation:

$$\begin{aligned} Pr[Y = i] &= \sum_j Pr[Y = i | X = j] \cdot Pr[X = j] \\ &= \sum_j \hat{A}_{ij} p_j \\ &= (\hat{A}\vec{p})_i \end{aligned}$$

A similar argument shows that $\hat{A}^t$ is the transition matrix of the Markov chain defined by random walks of length t; i.e. $(\hat{A}^t)_{ij}$ is the probability a random walk starting at i reached j in exactly t steps.
Clearly $\hat{A}\vec{u} = \vec{u}$ therefore the following theorem holds:

Theorem 3. *The stationary distribution of the random walk on G is the uniform distribution.*

The main result of this section is the following:

Theorem 4. $||\hat{A}^t\vec{p} - \vec{u}||_1 \le \sqrt{n} \cdot \alpha^t$ *for any distribution vector* $\vec{p}$.

In other words theorem 4 states that it doesn't matter what the initial distribution of the random walk is (it might be concentrated in one vertex), if $\alpha < 1$ we need to take only a logarithmic number of steps to get a distribution which is close to the uniform up to a polynomial factor.

Proof. Since $\hat{A}$ is symmetric, it has an orthonormal base $(\vec{\hat{v}_0}, \vec{\hat{v}_1}, \dots, \vec{\hat{v}_n})$.

Decompose $\vec{p}$ into the sum of the uniform distribution $\vec{u}$ and an error vector $\vec{\epsilon} = \vec{p} - \vec{u}$. The sum of the coordinates of $\vec{p}$ is 1, and the same is true for $\vec{u}$ therefore we have that the sum of the coordinates of $\vec{\epsilon}$ is 0. This means that $\langle \vec{u}, \vec{\epsilon} \rangle = 0$. In other words $\vec{\epsilon}$ is spanned by $(\vec{\hat{v}_1}, \dots, \vec{\hat{v}_n})$. So we have:

$$\begin{aligned} \hat{A}\vec{p} - \vec{u} &= \hat{A}(\vec{u} + \vec{\epsilon}) - \vec{u} \\ &= \hat{A}\vec{u} + \hat{A}\vec{\epsilon} - \vec{u} \\ &= \vec{u} + \hat{A}\vec{\epsilon} - \vec{u} \\ &= \hat{A}\vec{\epsilon} \end{aligned}$$

Therefore we have:

$$\begin{aligned} ||\hat{A}\vec{p} - \vec{u}||_2 &= ||\hat{A}\vec{\epsilon}||_2 \\ &\le \alpha ||\vec{\epsilon}||_2 \\ &\le \alpha ||\vec{p}||_2 \end{aligned}$$

where in the first inequality we used the fact that $\vec{c}$ is spanned by an orthonormal set of vectors for which the largest eigenvalue is α. We deduce that

$$||\hat{A}^t \vec{p} - \vec{u}||_2 \le \alpha^t$$

and therefore that

$$||\hat{A}^t \vec{p} - \vec{u}||_1 \le \sqrt{n} \cdot \alpha^t$$

□

3 A random walk on an expander yields a sequence of "good samples"

Consider the following problem: In an (n, d, α)-graph G there is a large set of good vertices (satisfying some condition) and we wish to find one of them. Let $B \subseteq V$ be the set of bad vertices, and assume that $\frac{|B|}{n} = \beta$. Let $x_1, \ldots, x_l$ be l vertices chosen uniformly at random from V, then $Pr[\forall i\ x_i \in B] \le \beta^l$. This approach uses $l \log n$ random bits. We will show that by choosing one vertex randomly, and then performing a random walk in G of length l, the probability that the random walk is confined to B is exponentially small in l.
First some intuition. Recall the expander mixing lemma, proven in the previous lecture:

Theorem 5 (Expander Mixing Lemma). *Let G be an (n, d, α)-graph, then for every $S, T \subseteq V(G)$ it holds that*

$$\left| \frac{d|S||T|}{n} - E(S,T) \right| \le \alpha d \sqrt{|S||T|} \le \alpha d n$$

Previously the expander mixing lemma was interpreted as saying that the number of edges between any two sets of vertices is not far from the expected for sets of those sizes. Now divide the inequality by dn to receive the following inequality:

$$\left| \frac{|S||T|}{n^2} - \frac{E(S,T)}{dn} \right| \le \alpha \tag{1}$$

Consider the following test: select uniformly at random (i, j) two vertices of G. Check whether $i \in S$ and $j \in T$. The term $\frac{|S||T|}{n^2}$ can be interpreted as the probability that this test succeeds.
Now consider a different test: select uniformly at random (i, j) an **edge** in G. Check whether $i \in S$ and $j \in T$. The term $\frac{E(S,T)}{dn}$ can be interpreted as the probability that this test succeeds. Note that the size of the probability domain of the first test is n^2 while the size of the probability domain of the second test is only nd, yet the difference between success probabilities is only a small constant α. In other words a random walk of length 1 can be viewed as discrepancy sets over sets of two vertices. Next we will show that random walks of length t are in fact discrepancy sets for sets of t vertices.
Let G be an (n, d, α)-graph, and $B \subset V$ with density $\beta = \frac{|B|}{|V|}$. Choose $X_0 \in_R V$ uniformly at random and let $X_0, \ldots, X_t$ be a random walk on G starting at X_0. Denote by (B, t) the event that the random walk is confined to B; i.e. that $\forall i\ X_i \in B$.

Theorem 6. $Pr[(B, t)] \le (\beta + \alpha)^t$

Let $P = P_B$ be a projection on the space of vectors supported in B, i.e.

$$P_{ij} = \begin{cases} 1 & \text{if } i = j \in B \\ 0 & \text{otherwise} \end{cases}$$

If v is a distribution vector, then Pv is the residual distribution vector of the distribution v conditioned on being in the set B. We need two lemmas:

Lemma 7. $Pr[(B, t)] = ||(P\hat{A})^t P\vec{u}||_1$

Proof. The action of P over a probability vector $\vec{v}$ is to nullify all the coordinated outside the set B, this transforms a probability vector into the residual probability vector of the same distribution, but conditioned on being in B. Thus $P\vec{u}$ is the residual probability of the uniform distribution conditioned to be in B. $\hat{A}P\vec{u}$ is the residual distribution after a random step has been taken. $P\hat{A}P\vec{u}$ is the residual probability conditioned on the random step remaining in B. Repeating this we see that $(P\hat{A})^t P\vec{u}$ is the residual probability vector of the random initial point and all t steps being in B. Since we don't care where in B we end up, we need to sum the coordinates of this vector, hence $Pr[(B,T)]$ is indeed given by $||(P\hat{A})^t P\vec{u}||_1$. □

Lemma 8. *For any non negative vector v:*

$$||P\hat{A}P\vec{v}||_2 \leq (\beta + \alpha) \cdot ||\vec{v}||_2$$

Proof. The idea of the proof is that $\hat{A}$ shrinks all components of a vector except the uniform distribution component, whereas P shrinks the uniform component without increasing anything else. Together they reduce all parts of the vector. Decompose $P\vec{v}$ into $P\vec{v} = (P\vec{v})_{||} + (P\vec{v})_{\perp}$ where $(P\vec{v})_{||}$ is the projection of $P\vec{v}$ on $\vec{u}$ and $\langle (P\vec{v})_{||}, (P\vec{v})_{\perp} \rangle = 0$. By the triangle inequality we know that

$$||P\hat{A}P\vec{v}||_2 \leq ||P\hat{A}(P\vec{v})_{||}||_2 + ||P\hat{A}(P\vec{v})_{\perp}||_2$$

First we look how P affects $(P\vec{v})_{||}$. Since $\vec{u}$ is an eigenvector of $\hat{A}$ with eigenvalue 1 we know:

$$||P\hat{A}(P\vec{v})_{||}||_2 = ||P(P\vec{v})_{||}||_2 = \sqrt{\beta} \cdot ||(P\vec{v})_{||}||_2$$

where the second equality is true since $(P\vec{v})_{||}$ is of the form $(a, a, \ldots, a)$ where a is some scalar.
If we fix $||v||_2$ then $||(P\vec{v})_{||}||_2$ is maximized when $\langle v, \vec{u} \rangle$ is maximized. In other words $||(P\vec{v})_{||}||_2$ is maximized when $\vec{v}$ is some scalar multiple of $\vec{u}$. Therefore

$$||P\hat{A}(P\vec{v})_{||}||_2 \leq \beta ||\vec{v}||_2.$$

Next we look at the effect on $(P\vec{v})_{\perp}$. P's effect is to multiply some coordinates by 0 without changing the others, so P can only shrink a vector. Since $(P\vec{v})_{\perp}$ is perpendicular to $\vec{u}$, it is spanned by the remaining eigenvectors of $\hat{A}$, all with eigenvalues at most α. This implies that

$$||P\hat{A}(P\vec{v})_{\perp}||_2 \leq ||\hat{A}(P\vec{v})_{\perp}||_2 \leq \alpha ||(P\vec{v})_{\perp}||_2$$

We note that $||(P\vec{v})_{\perp}||_2 \leq ||\vec{v}||_2$ and conclude that $||P\hat{A}(P\vec{v})_{\perp}||_2 \leq \alpha ||\vec{v}||_2$. Adding the two parts together we conclude that:

$$||P\hat{A}P\vec{v}||_2 \leq (\beta + \alpha) \cdot ||\vec{v}||_2$$

□

Now we use the lemma to prove theorem 6:

Proof. (theorem 6)

$$\begin{aligned} ||(P\hat{A})^t P\vec{u}||_1 &\leq \sqrt{n} \cdot ||(P\hat{A})^t P\vec{u}||_2 \\ &= \sqrt{n} \cdot ||(P\hat{A}P)^t \vec{u}||_2 \\ &\leq \sqrt{n} \cdot (\beta + \alpha)^t ||\vec{u}||_2 \\ &= (\beta + \alpha)^t \end{aligned}$$

□

3.1 Application: amplifying the success probability of random algorithms

Let L be some language in RP and assume that A is a randomized algorithm that decides whether $x \in L$ with a one sided error. Assume that A tosses m coins and has an error probability of β. Build an (n, d, α)-graph such that $V = \{0, 1\}^m$; i.e. the vertex set of the graph is the probability domain of A's coin tosses. Fix some input x and let B be all the coin tosses for which $A(x)$ is wrong. Now let A' be the following algorithm:

1. pick a vertex $v_0 \in V$ uniformly and at random.
2. perform a random walk of length t resulting with the set of vertices $(v_0, v_1, \ldots, v_t)$.
3. return $\bigcup_{i=0}^{t} A(x, v_i)$

A direct implication of theorem 6 yields that

$$Pr[A' \text{ fails }] = Pr[\forall i\ v_i \in B] \leq (\beta + \alpha)^t$$

The error probability is reduced exponentially while the number of random bits used is only $m + t \log d = m + O(t)$. Next we will show that the same trick can amplify the success probability of a two-sided error algorithm. In order to show this we need to restate theorem 6 in a stronger version.

Theorem 9. *Let $B_0, B_1, \ldots, B_t$ be subsets of V such that $\frac{|B_i|}{n} = \beta_i$. Define (B, t) to be the event that a random walk $(X_0, X_1, \ldots, X_t)$ has the property that $\forall i\ \ X_i \in B_i$. It holds that*

$$Pr[(B, t)] \leq \prod_{i=0}^{t-1} (\sqrt{\beta_i \beta_{i+1}} + \alpha)$$

Note that one should be able to strengthen that by a factor of $\sim \sqrt{\beta_0 \beta_t}$, but our simple argument used in the proof of Lemma 8 seems to lose that.

The proof of theorem 9 is indeed similar to the proof of theorem 6. Let P_i be the projection matrix corresponding to the set B_i. Lemma 7 should be restated such that

$$Pr[(B, t)] = \|\prod_{i=1}^{t} (P_i \hat{A}) P_0 \vec{u}\|_1$$

The analouge of Lemma 8 is

$$\|P_{i+1} \hat{A} P_i \vec{v}\|_{\leq} (\sqrt{\beta_i \beta_{i+1}} + \alpha) \|v\|,$$

and therefore theorem 9 follows.

Now let L be a language in BPP and assume that A is a randomized algorithms that decides whether $x \in L$ with a two sided error probability of $\beta \leq \frac{1}{10}$. As before assume that A tosses m coins and build an (n, d, α)-graph such that $V = \{0, 1\}^m$; i.e. the vertex set of the graph is the probability domain of A's coin tosses. Fix some input x and let B be all the coin tosses for which $A(x)$ is wrong. Now let A' be the following algorithm:

1. pick a vertex $v_0 \in V$ uniformly and at random.
2. perform a random walk of length t resulting with the set of vertices $(v_0, v_1, \ldots, v_t)$.
3. return $majority\{A(x, v_i)\}$

A' fails iff a majority of the v_i's are in B. Fix a set of indices $K \subset [t]$ such that $|K| \geq \frac{t}{2}$. For each $i \in K$ let $B_i = B$. We deduce from Theorem 6 that

$$Pr[\forall i \in K\ v_i \in B] \leq (\beta + \alpha)^{|K|} \leq (\beta + \alpha)^{\frac{t}{2}}$$

(note that sometimes the walk makes more that one transition before testing for membership in B). By assuming that α is small enough such that $\alpha + \beta \leq \frac{1}{8}$ and applying the union bound we deduce that:

$$Pr[A' \text{ fails }] \leq 2^t \cdot (\beta + \alpha)^{\frac{t}{2}} \leq 2^t \cdot \left(\frac{1}{8}\right)^{\frac{t}{2}} = \left(\frac{1}{2}\right)^{\frac{t}{2}}$$

We achieve an exponential reduction in the error probability using only $m + O(t)$ random bits. The following table sums up the parameters of the techniques presented for error reduction:

Method	Error Probability	No. of random bits
random algorithm A	$\frac{1}{10}$	m
t independent repetitions of A	2^{-t}	$t \cdot m$
Sampling a point and it's neighbors in an $(n, t, \frac{1}{\sqrt{t}})$-graph.	$\frac{1}{t}$	m
A random walk of length t on an $(n, d, \frac{1}{40})$-graph	$2^{-\frac{t}{2}}$	$m + O(t)$

4 Using expanders for hardness of approximation.

In this section we show another application for random walks on expanders. We will show that we can use such walks in order to establish a hardness of approximation result for an $\mathbf{NP}$ optimization problem - the maximum clique problem.

For a graph G, we define $\omega(G)$ to be the *clique number* of G. That is, $\omega(G)$ is the size of the maximum set $S \subseteq V(G)$ such that all vertices in S are neighbors of each other. Computing exactly the clique number of a graph is $\mathbf{NP}$-hard. Using the $\mathbf{PCP}$ Theorem, it is possible to prove that it is even hard to *approximate* the clique number to within a constant factor. That is, we have the following theorem:

Theorem 10. *There exists a number $0 < a < 1$, such that it is NP-hard to distinguish between the following cases:*

1. $\omega(G) \leq an$

   and

2. $\omega(G) \geq 1.1an$

In this section we will show that even obtaining a *very rough* approximation for $\omega(G)$ is $\mathbf{NP}$-hard. That is, we will show that it is $\mathbf{NP}$-hard to approximate $\omega(G)$ even within a factor of n^ϵ for some $\epsilon > 0$. That is, we will prove the following theorem:

Theorem 11. *There exists a number $\epsilon > 0$, such that if there is a polynomial-time algorithm A such that for every graph G with n vertices*

$$n^{-\epsilon} \leq \frac{A(G)}{\omega(G)} \leq n^\epsilon$$

then $\mathbf{NP} = \mathbf{P}$.

Note: The results of this section have been superseded by a result of Håstad that it is $\mathbf{NP}$-hard to approximate $\omega(G)$ even within a factor $n^{1-\alpha}$ for *every* $\alpha > 0$. However, our approach will involve simpler analysis (and of course, expander graphs).

4.1 Proof of Theorem 11.

To illustrate the main ideas behind the proof of Theorem 11, we will prove a weaker version of this theorem. In the weak version we will prove under the same assumption the weaker conclusion that $\mathbf{NP} \subseteq \mathbf{RP}$ (instead of $\mathbf{NP} = \mathbf{P}$).

Lemma 12 (Theorem 11, weak version). *There exists a number $\epsilon > 0$, such that if there is a polynomial-time algorithm A such that for every graph G with n vertices*

$$n^{-\epsilon} \leq \frac{A(G)}{\omega(G)} \leq n^\epsilon$$

then $\mathbf{NP} \subseteq \mathbf{RP}$.

After we prove Lemma12, we will use the ideas of the proof, along with random walks on expanders to obtain Theorem 11, which can be looked at as a derandomized version of Lemma 12.

4.1.1 Proof of Lemma 12

(Since this proof will be superseded by the proof of Theorem 11, we allow ourselves some slackness.)

We will let ϵ be some constant, whose value will be determined later. Suppose that there exists a polynomial-time algorithm A that distinguishes between the two cases of Theorem 11. We will show that there exists a probabilistic polynomial-time algorithm B to distinguish between the two cases of Theorem 10, thus showing that $\mathbf{NP} \subseteq \mathbf{RP}$.[1]

Our algorithm B will work as follows:

Algorithm B

- Input: A graph G on n vertices.

1. Construct the graph H, where H is the following t-th power of G, for $t = \Theta(\log n)$: The vertex set $V(H)$ is the set V^t of all t-tuples in V. The edge set $E(H)$ is defined as follows: $\langle (v_1, \ldots, v_t), (u_1, \ldots, u_t) \rangle$ is an edge in $E(H)$ iff the set $\{v_1, \ldots, v_t\} \cup \{u_1, \ldots, u_t\}$ is a clique in G.
2. Let H' be the graph obtained from H by sampling $m = \Theta(a^{-t})$ $(= n^{\Theta(1)})$ vertices from H at random and taking their induced graph.
3. Return 1 if $A(H') > n^{\epsilon}$ and 0 otherwise. (ϵ will be determined later.)

It may seem like Algorithm B runs in time $n^{O(\log n)}$ instead of polynomial-time because the size of the graph H will be $n^{O(\log n)}$. However the construction of H in Step 1 can be done *implicitly* (that is, we don't to write out the full graph H) and so Algorithm B can be implemented in probabilistic polynomial-time.

We have the following claim:

Claim 12.1. $\omega(H) = \omega(G)^t$

Proof. Clearly if S is a clique in G then the set S^t is a clique. Therefore $\omega(H) \geq \omega(G)^t$.

On the other hand we claim that $\omega(H) \leq \omega(G)^t$. Indeed, if S' is a clique in H then the union of all tuples in S' is a clique in G. If $|S'| > k^t$ then it must be that this union contains more than k elements. ☐

For every clique $S \subseteq V(H)$, the expected fraction of vertices in H' that are in S is $\frac{|S|}{|V|^t}$. With high probability we will have that for *every* clique $S \subseteq V(H)$, the fraction of vertices in S chosen to be in H' is $\Theta(\frac{|S|}{|V|^t})$. Therefore we have that with high probability we will have that $\omega(H') = \Theta(\frac{\omega(H)}{|V|^t} \cdot m) = \Theta(\frac{\omega(G)^t}{n^t} \cdot m)$. We see that:

1. If $\omega(G) \leq an$ then $\omega(H') \leq a^t m = \Theta(1)$.
2. If $\omega(G) \geq 1.1an$ then $\omega(H') \geq 1.1^t \Theta(1) \sim m^{2\epsilon}$ (for $\epsilon \sim \frac{\log 1.1}{-2 \log a}$).

We see that with high probability Algorithm B will return 1 if $\omega(G) > 1.1an$ and 0 if $\omega(G) < an$ which is what we wanted to prove.

4.1.2 The Actual Proof

Now that we have proved Lemma 12, we will now use the ideas of this proof, along with random walk on expander graphs, to prove Theorem 11. We will again let ϵ be some constant, whose value will be determined later, and assume that there exists a polynomial-time algorithm A that distinguishes between the two cases of Theorem 11. We will use A this time to show that there exists a *deterministic* polynomial-time algorithm B' to distinguish between the two cases of Theorem 10, thus showing that $\mathbf{NP} = \mathbf{P}$.

The only difference between Algorithm B' and Algorithm B, described in Section 4.1.1, is that in Step 2, Algorithm B' will use a *derandomized sampling* to construct the graph H'. The sampling will work in the following way. We will construct a (n, d, α)-expander $\mathcal{G}$ such that $V(\mathcal{G}) = V(G)$. We will then choose the set of t-tuples to be sampled in H' as the set of all t-tuples that represent a *length t walk* in the graph $\mathcal{G}$. We again let m denote the number of

[1] It may seem as if we only show that $\mathbf{NP} \subseteq \mathbf{BPP}$ but it is not hard to show (using the self-reducibility of $\mathbf{NP}$-complete problems) that if $\mathbf{NP} \subseteq \mathbf{BPP}$ then $\mathbf{NP} \subseteq \mathbf{RP}$.

vertices in H', note that $m = nd^{t-1}$ (where d is the degree of $\mathcal{G}$). If $t = \Theta(\log n)$ and d is constant then this value is polynomial in n. What we have already seen is that a random length t walk in $\mathcal{G}$ does sometimes behave similarly to a random t-tuple. We need to show that this holds also in this context.

We start with the following claim:

Claim 13. *Suppose that $\omega(G) \leq an$. For every clique $S \subseteq V(G)$ in G, the probability that a length t random walk in $\mathcal{G}$ doesn't leave S (i.e., is contained in S^t) is at most $(a + \alpha)^t$.*

Proof. This is a direct application of Theorem 6. □

As a corollary we obtain the following:

Corollary 14. *If $\omega(G) \leq an$ then $\omega(H') \leq (a + \alpha)^t m$*

Proof. A set $U \subseteq V(H')$ is a clique in H' if and only if all tuples in U are part of the same clique in G. Since we assume that $\omega(G) \leq an$, this means that $\frac{|U|}{m}$ can be at most $(a + \alpha)^t$. □

For the other direction, we need to prove the following claim:

Claim 15. *Suppose that $\omega(G) \geq 1.1an$. Let $S \subseteq V(G)$ be a maximum sized clique in G (i.e., $|S| \geq 1.1an$). The probability that a length t random walk in $\mathcal{G}$ doesn't leave S (i.e., is contained in S^t) is at* least $(1.1a - 2\alpha)^t$.

Proof. This is an application of the following theorem that is analogous to Theorem 6:

Theorem 16. *In the notation of Theorem 6, suppose that $\beta > 6\alpha$. Then,*

$$Pr[(B, t)] \geq (\beta - 2\alpha)^t$$

Theorem 16 provides a lower bound on the probability that a random walk does not leave a specified set of fraction β. It shows that this probability is not much smaller than than β^t (which is what happens if we choose t independent random vertices). The proofs of Theorem 6, 16 can be found in the paper "Derandomized Graph Products" by Alon, Feige, Wigderson and Zuckerman.[2] We remark that an analogous theorem to Theorem 9 also holds (i.e., a lower bound on the probability to stay in *changing* sets).

□

We now have the following corollary:

Corollary 17. *If $\omega(G) \geq 1.1an$ then $\omega(H') \geq (1.1a - \alpha)^t m$.*

Using both corollaries we see that if we choose α small enough such that $\alpha < 1 - a$ and $\alpha < a/30$ (we can take the graph $\mathcal{G}^c$ for some constant c to ensure this) then we get that

1. If $\omega(G) \leq an$ then $\omega(H') \leq \beta^t m$ for some constant $\beta < 1$.
2. If $\omega(G) \geq 1.1an$ then $\omega(H') \geq \gamma^t m$ for some constant $\gamma > \beta$.

Since $(\gamma/\beta)^t = n^{\epsilon'} = m^{2\epsilon}$ for some constants ϵ', ϵ we see that we can use A to distinguish between the two cases.

[2] Available from Avi Wigderson's homepage on `http://www.math.ias.edu/~avi/PUBLICATIONS/`.