A Sublinear Tolerant Max Cut Tester for Bounded Degree Expander Graphs

Edin Guso
Supervised by: Michael Kapralov and Akash Kumar

Abstract

We present a sublinear-time algorithm for testing whether a bounded degree expander graph has a max-cut value close to 1 or far from 1. Graphs are represented by incidence lists of bounded length d, and the testing algorithm can perform queries of the form: "who is the i-th neighbor of vertex v". The tester should determine with high probability whether the max-cut value is greater than $1 - \epsilon_1$ or less than $1 - \epsilon_2$. Our testing algorithm has query complexity and running time $O(\sqrt{n}\log^2 n/\epsilon_1)$ where n is the number of graph vertices and ϵ_1 is the tolerance value.

1 Introduction

Property testers are commonly used in graph theory and can be split into two categories based on the representations of graph being used. Each model requires different techniques and involves different complexities [5]. For dense graphs, usually the adjacency-matrix (adjacency-predicate) representation is used [3]. For bounded-degree graphs on the other hand, incidence-list representation is used [4].

Property testers accept instances that have the given property and reject instances that are far from having that property. Tolerant property testers can additionally accept instances that are close to having the given property. Recently, a state-of-the-art tolerant bipartiteness tester for dense graphs has been published [2]. Our work is focused on developing a tolerant max-cut tester for the bounded-degree model.

The foundation of this project is based on Goldreich and Ron's bipartiteness tester on bounded degree graphs. Goldreich and Ron have developed a bipartiteness tester with complexity $O(\sqrt{n} \cdot poly(\log n/\epsilon))$ that always accepts bipartite graphs and rejects graphs that are ϵ-far from being bipartite with probability at least 2/3 (for constant ϵ) [5].

Our main contribution to Goldreich and Ron's paper is that our algorithm is a tolerant tester. Our analysis shows that `TolerantMaxCut` (1) algorithm with complexity $O(\sqrt{n}\log^2 n/\epsilon_1)$ not only rejects graphs with cut value smaller than $1 - \epsilon_2$ (graphs that are ϵ_2-far from being bipartite) with probability at least 2/3, but also accepts graphs with cut value larger than $1 - \epsilon_1$ (graphs that are ϵ_1-close to being bipartite) with probability at least 2/3 (for constant ϵ_2 and $\epsilon_1 = O(1/\log^4 n)$.

2 Preliminaries and Definitions

Property Testers: A property testing algorithm $\mathscr{A}$ is required to accept objects that have the property $\mathscr{P}$ and reject objects that are ϵ-far from having $\mathscr{P}$, for some parameter $0 \leq \epsilon \leq 1$ with high probability.

Tolerant Property Testers: A tolerant property testing algorithm $\mathscr{A}$ is required to accept objects that are ϵ_1-close to having that property $\mathscr{P}$ and reject objects that are ϵ_2-far from having $\mathscr{P}$, for some parameters $0 \leq \epsilon_1 < \epsilon_2 \leq 1$ with high probability [6].

Max-Cut of a Graph: For a graph $G = (V, E)$, a maximum cut is a cut whose size is at least the size of any other cut. That is, it is a partition of the graph's vertices into two complementary sets S and $V \backslash S$, such that the fraction of edges between the set S and the set $V \backslash S$ is as large as possible. We let $Cut(G)$ denote the max cut of the graph G.

Expansion of a Graph: Let $G = (V, E)$ be a graph and $S \subseteq V$ be a vertex set. Let $\sigma(S)$ denote the sum of degrees of all vertices $v \in S$, that is $\sigma(S) = \sum_{v \in S} d(v)$. We say that an edge $e \in E$ is cut by S if $e \cap S \neq \emptyset$ and $e \cap (V \backslash S) \neq \emptyset$. In this context, we often call a pair $(S, V \backslash S)$ a cut. Let $E(S, V \backslash S)$ denote the set of edges cut by S. Then, the expansion of S is:

$$\Phi(S) = \Phi(V \backslash S) = \frac{|E(S, V \backslash S)|}{min\{\sigma(S), \sigma(V \backslash S)\}}$$

The expansion of a graph $G = (V, E)$ is defined to be $\Phi(G) := min_{S \subseteq V} \Phi(S)$. For $\Phi > 0$, we say that G is a Φ-expander if $\Phi(G) \geq \Phi$.

Lazy Random Walks: The algorithm provided in the next section relies on random walks on graphs. However, the traditional random walk is a periodic Markov chain if the graph is bipartite. Therefore, instead of traditional random walks, lazy random walks are used to make the Markov chain aperiodic. Let $p_{s,v}(1) \equiv p_{s,v}$ denote the probability of a random walk moving from any vertex s to any vertex v in a single step. The transition probabilities of lazy random walks on any graph $G = (V, E)$ are assigned as follows:

$$\forall s \in V \quad p_{s,s} = 1 - \frac{|N(s)|}{2d}$$
$$\forall s \in V \quad \forall v \in N(s) \quad p_{s,v} = \frac{1}{2d}$$
$$\forall s \in V \quad \forall v \notin N(v) \quad p_{s,v} = 0$$

where $d = \Delta(G)$, $\Delta(G)$ denotes the maximum degree of the graph G and $N(s)$ denotes the neighbourhood of vertex v. These random walks are called lazy because there is at least $1/2$ chance that the walk will stay at the same vertex since $|N(s)| \leq d$, $\forall s \in V$.

Set of Lazy Random Walks: We define $W_s(l)$ to be the set of all random walks starting at a fixed vertex s of length l. Walks sampled uniformly at random from this set are denoted as $w \sim W_s(l)$.

Effective Length of Lazy Random Walks: When considering the effective length of random walks and classifying walks as odd or even, we will not be counting the self-loops in the random walks. Therefore, even if all the walks will have the same length, they will end up having different effective lengths due to the self-loops.

Lazy Random Walk Probabilities: Let $p_{s,v}^{even}(l)$ denote the probability that a random walk of length l with an even effective length that starts at a fixed vertex s will end at a fixed vertex v. Similarly, let $p_{s,v}^{odd}(l)$ denote the probability that a random walk of length l with an odd effective length that starts at a fixed vertex s will end at a fixed vertex v. Finally, $p_{s,v}(l) = p_{s,v}^{even}(l) + p_{s,v}^{odd}(l)$.

Mixing Time of Lazy Random Walks: Lazy random walks mix in $c \log n$ steps if G is an $\Omega(1)$-expander ($\Phi(G) = \Omega(1)$) where $c > 1$ is a constant. This means that the walks reach stationary distribution in that many steps.

Fact. *Let l^* be the necessary number of steps to reach the stationary distribution in a graph $G = (V, E)$. Then:*

$$\forall s, v \in V \quad \frac{1}{2n} \leq p_{s,v}(l^*) \leq \frac{2}{n} \tag{1}$$

where $n = |V|$.

Partition of a Graph: For any vertex $s \in V$ that we fix, every vertex $v \in V$ is automatically assigned two probabilities: $p_{s,v}^{even}(l)$ and $p_{s,v}^{odd}(l)$. We can divide these vertices into two sets, based on whether it is more likely to reach them by an even or odd effective lazy random walk length. In a bipartite graph, this would result in exactly the bipartition of the graph since every vertex is reachable by either an even or odd length walk. Therefore, we say that whenever we fix a vertex $s \in V$, the graph is automatically partitioned into $V = V_s^{even} \cup V_s^{odd}$:

$$V_s^{even} = \{v \in V | p_{s,v}^{even}(l) \geq p_{s,v}^{odd}(l)\}$$
$$V_s^{odd} = \{v \in V | p_{s,v}^{odd}(l) > p_{s,v}^{even}(l)\}$$

Violating Edges w.r.t a Partition: In a graph $G = (V, E)$ for a fixed vertex $s \in V$, an edge $e \in E$ is a violating edge if $e \cap V_s^{even} \neq \emptyset$ and $e \cap V_s^{odd} \neq \emptyset$. We also let E_{viol} denote the set of violating edges. We call these edges 'violating' because if we remove all violating edges from the graph, we would have $\mathcal{C}ut(G) = 1$ (i.e. G would be bipartite).

Main Idea: Similar to Goldreich and Ron's paper, we are interested in detecting odd cycles [5]. Because the number of odd cycles in the graph is related to how far the graph is from being bipartite. We develop our analysis around the following quantity:

$$r_s = \sum_{v \in V} p_{s,v}^{even}(l) p_{s,v}^{odd}(l)$$

This summation was also used in Goldreich and Ron's paper. r_s is key in our analysis because if the graph is bipartite, either $p_{s,v}^{even}(l) = 0$ or $p_{s,v}^{odd}(l) = 0$ $\forall s, v \in V$. Therefore $r_s = 0$ $\forall s \in V$ if $G = (V, E)$ is bipartite. We will further show that r_s can be upper bounded for the ϵ_1-close case and lower bounded for the ϵ_2-far case. The gap between these bounds will be utilized while deciding whether a graph is close or far. Check sections 4 and 5 for proofs.

3 Algorithm

The `TolerantMaxCut` (1) algorithm is a tolerant max-cut tester for d-regular $\Omega(1)$-expander graphs. The algorithm picks a set of vertices $S \subseteq V$ uniformly at random such that $|S| = 1/\epsilon_1$. Then, for every vertex $s \in S$, `CountOddCycle` (2) procedure is called on the vertex s. `CountOddCycle` (2) procedure counts the number of encountered odd cycles. If the returned value is larger than $\epsilon_2 \log^2 n/256$, then the graph is most likely far from being bipartite (i.e. it has a small max-cut value). We increment the count c for each such vertex. After all the vertices $s \in S$ have been processed, we look at the value of c. If c is smaller than $1/2\epsilon_1$ (i.e. if at most half the vertices incremented the count), we accept and otherwise, we reject.

Algorithm 1 TolerantMaxCut

```
G = (V, E) is the input graph and ε1 and ε2 are the tolerant max cut parameters
Pick S ⊆ V : |S| = 1/ε1
c ← 0
for s ∈ S do
    Tot(s) ← CountOddCycle(s)
    if Tot(s) ≥ ε2 log^2 n / 256 then
        c+ = 1
    end if
end for
if c < 1/(2ε1) then
    Accept
else
    Reject
end if
```

The `CountOddCycle` (2) procedure takes a vertex $s \in V$ as input and performs $\sqrt{n} \log n$ random walks of length $l = O(\log n)$ starting from s. The number of random walks is in the order of $\sqrt{n}$ in order to have collisions as shown by the birthday paradox. Each collision creates a cycle, and the number of odd cycles encountered provides us information about how far this graph is from being bipartite. After each completed random walk the vertex is added to the set B. Also, if the effective length of the walk was even, the vertex is incremented in dictionary B_{even}, otherwise it is incremented in dictionary B_{odd}. For example, if a vertex v was reached twice with an even length walk, then $B_{even}[v] = 2$. Finally, the number of odd cycles encountered is computed and returned.

Theorem. *The algorithm `TolerantMaxCut` (1) constitutes a tolerant tester for cut value of a d-regular $\Omega(1)$-expander graph G with complexity $O(\sqrt{n} \log^2 n/\epsilon_1)$, where ϵ_2 is a constant and $\epsilon_1 = O(1/\log^4 n)$. Specifically:*

- *If $\mathcal{C}ut(G) \geq 1 - \epsilon_1$ then the algorithm accepts with probability at least $2/3$.*
- *If $\mathcal{C}ut(G) \leq 1 - \epsilon_2$ then the algorithm rejects with probability at least $2/3$.*

Algorithm 2 CountOddCycle

s is the input start vertex and ϵ_1 and ϵ_2 are the tolerant max cut parameters
$B \leftarrow \emptyset$
$B_{odd} \leftarrow \{\}$
$B_{even} \leftarrow \{\}$
for $t \leftarrow 1$ to $\sqrt{n}\log n$ **do**
  $v \leftarrow$ Perform an $l = O(\log n)$ length lazy random walk from s
  $B \leftarrow B \cup \{v\}$
  if v is reached by an even effective length **then**
    $B_{even}[v] = B_{even}[v] + 1$
  else
    $B_{odd}[v] = B_{odd}[v] + 1$
  end if
end for
Return $\sum_{v \in B} |B_{even}(v)| B_{odd}(v)|$

4 Case 1: $\mathcal{C}ut(G) \geq 1 - \epsilon_1$

Lemma 1. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, then, for any $0 < \alpha < 1$, for at least $(1 - \alpha)n$ vertices $s \in V$ it holds that:*

$$r_s \leq \frac{8\alpha}{n} \tag{2}$$

Definition 1.1. In a graph $G = (V, E)$, fix $s \in V$ and $w \in W_s(l)$. Then, we define $Y_{s,w}$ to be the number of violating edges encountered during the walk w starting at vertex s. We let:

$$Y_s = \mathbb{E}_{w \sim W_s(l)}[Y_{s,w}]$$

where Y_s is the expectation of $Y_{s,w}$'s over a uniformly random walk. We also let:

$$Y = \mathbb{E}_{s \sim V}[Y_s]$$

where Y is the expectation of Y_s's over a uniformly random start vertex.

Claim 1.1. *If $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, then:*

$$Y \leq l\epsilon_1 \tag{3}$$

Proof of Claim 1.1. By definition of Y, we have that:

$$Y = \mathbb{E}_{s \sim V}[Y_s] = \mathbb{E}_{s \sim V}[\mathbb{E}_{w \sim W_s(l)}[Y_{s,w}]] = \mathbb{E}_{s \sim V, w \sim W_s(l)}[Y_{s,w}]$$

We let $\{e_1, e_2, \ldots, e_{l-1}\}$ denote the edges visited by $w_s(l)$. Note that, expected number of violating edges on an l-length walk is equal to the sum of expectations of each edge on the walk being a violating edge. Then:

$$\mathbb{E}_{s \sim V, w \sim W_s(l)}[Y_{s,w}] = \mathbb{E}_{s \sim V, w \sim W_s(l)}[\sum_{t=1}^{l-1} \mathbb{1}_{e_t \in E_{viol}}]$$

w starts at a stationary distribution because we picked a uniformly random vertex, and remains at the stationary distribution because we picked a uniformly random walk. Therefore $\mathbb{E}[\mathbb{1}_{e_t \in E_{viol}}] \leq \epsilon_1$. Then:

$$Y = \mathbb{E}_{s \sim V, w \sim W_s(l)}[\sum_{t=1}^{l-1} \mathbb{1}_{e_t \in E_{viol}}] \leq (l-1)\epsilon_1 \leq l\epsilon_1$$

$\square$

Corollary 1.1. *If $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, then, for any $0 < \alpha < 1$, for at least $(1 - \alpha)n$ vertices $s \in V$ it holds that:*

$$Y_s < \frac{1}{\alpha} l \epsilon_1 \tag{4}$$

Proof of Corollary 1.1. We first apply Markov's inequality:

$$\mathbb{P}(Y_s \geq \frac{1}{\alpha}\mathbb{E}[Y_s]) \leq \alpha$$

Note that as a direct consequence of eq.3, we have that $\mathbb{E}_{s \sim V}[Y_s] \leq l\epsilon_1$. When we plug this in and take the complementary event, we get that:

$$\mathbb{P}(Y_s < \frac{1}{\alpha} l \epsilon_1) > 1 - \alpha$$

□

Definition 1.2 (α-Good Vertex)**.** We call a vertex $s \in V$ α-good if it satisfies eq.4. Therefore, Corollary 1.1 is equivalent to saying that if $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, then, for any $0 < \alpha < 1$, at least $(1 - \alpha)n$ vertices are α-good. We also define V_{good} to be the set of α-good vertices. Similarly, Lemma 1 is equivalent to saying that if $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, then, for any $0 < \alpha < 1$, any vertex $s \in V_{good}$ has that $r_s \leq 8\alpha/n$.

Corollary 1.2. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then, for $w \sim W_s(l)$, it holds that:*

$$\mathbb{P}(Y_{s,w} \geq \frac{1}{\alpha^2} l \epsilon_1) \leq \alpha \tag{5}$$

Proof of Corollary 1.2. We first apply Markov's inequality:

$$\mathbb{P}(Y_{s,w} \geq \frac{1}{\alpha}\mathbb{E}[Y_{s,w}]) \leq \alpha$$

Note that since s was picked to be an α-good vertex, as a direct consequence of eq.4, we have that $\mathbb{E}_{w \sim W_s(l)}[Y_{s,w}] \leq (1/\alpha) l \epsilon_1$. When we plug this in, we achieve the required result. □

Note. Summation r_s is over all vertices $v \in V$. In order to upper bound r_s, we can split the summation into smaller terms and upper bound individual terms. Namely, $V = V_{good} \cup V_{bad}$, where $V_{bad} = V \backslash V_{good}$. Furthermore, we define sets $S_{even} = V_{good} \cap V_{even}$ and $S_{odd} = V_{good} \cap V_{odd}$. Clearly, sets S_{even}, S_{odd} and V_{bad} are disjoint sets whose union is equal to V. Therefore, we can split the summation r_s among these sets.

Claim 1.2. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then:*

$$\sum_{v \in V_{bad}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) \leq \frac{4\alpha}{n} \tag{6}$$

Proof of Claim 1.2. By eq.1, we have that $p_{s,v}(l) < 2/n$. This implies that $p_{s,v}^{even}(l) < 2/n$ and $p_{s,v}^{odd}(l) < 2/n$. By substituting these, we have that:

$$\sum_{v \in V_{bad}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) \leq \sum_{v \in V_{bad}} \frac{4}{n^2}$$

From Definition 1.2 we have that the number of good vertices is at lease $(1 - \alpha)n$. This implies that number of bad vertices is at most αn. Using this fact:

$$\sum_{v \in V_{bad}} \frac{4}{n^2} \leq \alpha n \frac{4}{n^2} = \frac{4\alpha}{n}$$

□

Definition 1.3. We let $e_{s,v}^{even}(l)$ denote the event corresponding to the probability $p_{s,v}^{even}(l)$, namely $\mathbb{P}(e_{s,v}^{even}(l)) = p_{s,v}^{even}(l)$. Similarly $e_{s,v}^{odd}(l)$ is defined such that $\mathbb{P}(e_{s,v}^{odd}(l)) = p_{s,v}^{odd}(l)$. Additionally, we let $e_{s,v}^{viol}(l)$ denote the event that a random walk encounters violating edge.

Claim 1.3. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $Cut(G) \geq 1-\epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then:*

$$\sum_{v \in S_{even}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) \leq \frac{2\alpha}{n} \tag{7}$$

Proof of Claim 1.3. Using the definitions given above, we rephrase the quantity we are interested in:

$$\sum_{v \in S_{even}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) = \sum_{v \in S_{even}} p_{s,v}^{even}(l) \mathbb{P}(e_{s,v}^{odd}(l) \wedge \overline{e_{s,v}^{viol}(l)}) + \sum_{v \in S_{even}} p_{s,v}^{even}(l) \mathbb{P}(e_{s,v}^{odd}(l) \wedge e_{s,v}^{viol}(l))$$

The above equation holds because $p_{s,v}^{odd}(l) = \mathbb{P}(e_{s,v}^{odd}(l) \wedge \overline{e_{s,v}^{viol}(l)}) + \mathbb{P}(e_{s,v}^{odd}(l) \wedge e_{s,v}^{viol}(l))$. However, we show that:

$$\mathbb{P}(e_{s,v}^{odd}(l) \wedge \overline{e_{s,v}^{viol}(l)}) = 0$$

because moving from $s \in V_{even}$ to $v \in V_{even}$ with an odd length walk without encountering a violating edge is impossible. Therefore, the quantity simplifies to:

$$\sum_{v \in S_{even}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) = \sum_{v \in S_{even}} p_{s,v}^{even}(l) \mathbb{P}(e_{s,v}^{odd}(l) \wedge e_{s,v}^{viol}(l))$$

Using the conditional probability formula, we rephrase the quantity as:

$$\sum_{v \in S_{even}} p_{s,v}^{even}(l) \mathbb{P}(e_{s,v}^{odd}(l) \wedge e_{s,v}^{viol}(l)) = \sum_{v \in S_{even}} p_{s,v}^{even}(l) \mathbb{P}(e_{s,v}^{odd}(l) | e_{s,v}^{viol}(l)) \mathbb{P}(e_{s,v}^{viol}(l))$$

By eq.1, we have that $p_{s,v}(l) \leq 2/n$. This implies that $p_{s,v}^{even}(l) \leq 2/n$. Also, we note that:

$$\mathbb{P}(e_{s,v}^{viol}(l)) = \mathbb{P}(Y_{s,w} \geq \frac{1}{\alpha^2} l \epsilon_1)$$

as long as $0 < l\epsilon_1/\alpha^2 < 1$. We will ensure this remains true by the selection of α and ϵ_1 later in the analysis. Combining with eq.1.2, we have that:

$$\mathbb{P}(e_{s,v}^{viol}(l)) \leq \alpha$$

Now, when we substitute these observations in the quantity we are interested in, we get that:

$$\sum_{v \in S_{even}} p_{s,v}^{even}(l) \mathbb{P}(e_{s,v}^{odd}(l) \wedge e_{s,v}^{viol}(l)) \leq \sum_{v \in S_{even}} \frac{2}{n} \alpha \mathbb{P}(e_{s,v}^{odd}(l) | e_{s,v}^{viol}(l)) = \frac{2\alpha}{n} \sum_{v \in S_{even}} \mathbb{P}(e_{s,v}^{odd}(l) | e_{s,v}^{viol}(l))$$

We let $e_s^{odd}(l)$ denote the event that a random walk of length l with an even effective length that starts at a fixed vertex s will end at any vertex $v \in S_{even}$. Formally:

$$e_s^{odd}(l) = \bigcup_{v \in S_{even}} e_{s,v}^{odd}(l)$$

We notice that this is a disjoint union of events. Therefore:

$$\mathbb{P}(e_s^{odd}(l) | e_{s,v}^{viol}(l)) = \sum_{v \in S_{even}} \mathbb{P}(e_{s,v}^{odd}(l) | e_{s,v}^{viol}(l))$$

Since $\mathbb{P}(e_s^{odd}(l)|e_{s,v}^{viol}(l))$ is a probability, it cannot be larger than 1. Therefore, we can upper bound:

$$\sum_{v \in S_{even}} \mathbb{P}(e_{s,v}^{odd}(l)|e_{s,v}^{viol}(l)) \leq 1$$

When we put everything together, we get that:

$$\frac{2\alpha}{n} \sum_{v \in S_{even}} \mathbb{P}(e_{s,v}^{odd}(l)|e_{s,v}^{viol}(l)) \leq \frac{2\alpha}{n}$$

□

Corollary 1.3. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then:*

$$\sum_{v \in S_{odd}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) \leq \frac{2\alpha}{n} \tag{8}$$

Proof of Corollary 1.3. The proof follows exactly the same steps as the Proof of Claim 1.3. □

Proof of Lemma 1. We discussed in a previous note that the summation r_s can be split into smaller terms:

$$r_s = \sum_{v \in V} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) = \sum_{v \in V_{bad}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) + \sum_{v \in S_{even}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) + \sum_{v \in S_{odd}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l)$$

From eq.6, eq.7 and eq.8 we know that each of these terms can be individually upper bounded. Therefore, we achieve the required result:

$$r_s \leq \frac{4\alpha}{n} + \frac{2\alpha}{n} + \frac{2\alpha}{n} = \frac{8\alpha}{n}$$

□

Definition 1.4. The total number of odd cycles returned from `CountOddCycle` (2) is denoted as $Tot(s)$. We also define:

$$\eta_{ij} = \begin{cases} 1 & i\text{-th and } j\text{-th walks create an odd cycle} \\ 0 & \text{otherwise} \end{cases}$$

It is important to notice that:

$$Tot(s) = \sum_{1 \leq i < j \leq t} \eta_{ij}$$

This equation is important because it provides the intuition of the birthday paradox. Here, we have $t(t-1)/2$ pairs of walks and all these walks end up in one of n vertices. Therefore the expected number of collisions is $(t(t-1)/2)^2/n$. We will encounter this quantity throughout the analysis.

Lemma 2. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \geq 1-\epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$. Then the expected value that `CountOddCycle` (2) returns is:*

$$\mathbb{P}(Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}) \leq \frac{O(\alpha)}{\epsilon_2} \tag{9}$$

Definition 2.1 (Exotic Vertex)**.** A vertex $s \in V$ is said to be exotic with respect to `CountOddCycle` (2) if the returned value from the routine is large. Specifically, $s \in V$ is said to be exotic if:

$$Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}$$

Corollary 2.1. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $Cut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$. Then:*

$$\mathbb{P}(s \textit{ is an exotic vertex}) \leq \frac{O(\alpha)}{\epsilon_2} \tag{10}$$

Proof of Corollary 2.1. This result is a direct consequence of Lemma 2 □

Claim 2.1. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $Cut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then during the `CountOddCycle` (2) routine:*

$$\mathbb{E}[\eta_{ij}] \leq \frac{16\alpha}{n} \quad \forall i, j \in [t] \tag{11}$$

Proof of Claim 2.1. The probability of two different random walks creating an odd cycle is equal to two random walks ending at any same vertex with different parities:

$$\mathbb{E}[\eta_{ij}] = 2 \sum_{v \in V} p_{s,v}^{even}(l) p_{s,v}^{odd}(l)$$

The factor of two comes from the fact that either of the walks can be even. The summation is exactly equal to r_s. Therefore, using eq.2, we say that:

$$\mathbb{E}[\eta_{ij}] = 2r_s \leq \frac{16\alpha}{n}$$

□

Claim 2.2. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $Cut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then:*

$$\mathbb{E}[Tot(s)] \leq 16\alpha \log^2 n \tag{12}$$

Proof of Claim 2.2. We first use the fact that $Tot(s)$ is equal to summation of η_{ij}'s:

$$\mathbb{E}[Tot(s)] = \mathbb{E}[\sum_{1 \leq i < j \leq t} \eta_{ij}] = \sum_{1 \leq i < j \leq t} \mathbb{E}[\eta_{ij}]$$

We substitute the result from eq.11 and use the fact that there are less than t^2 pairs of walks:

$$\sum_{1 \leq i < j \leq t} \mathbb{E}[\eta_{ij}] \leq t^2 \frac{16\alpha}{n}$$

Now, we substitute the number of walks $t = \sqrt{n} \log n$:

$$\mathbb{E}[Tot(s)] \leq n \log^2 n \frac{16\alpha}{n} = 16\alpha \log^2 n$$

□

Corollary 2.2. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $Cut(G) \geq 1 - \epsilon_1$, fix $0 < \alpha < 1$ and let $s \in V$ be an α-good vertex. Then:*

$$\mathbb{P}(Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}) \leq \frac{4096\alpha}{\epsilon_2} \tag{13}$$

Proof of Corollary 2.2. We first apply Markov's inequality:

$$\mathbb{P}(Tot(s) \geq a) \leq \frac{\mathbb{E}[Tot(s)]}{a}$$

We then substitute $a = \epsilon_2 \log^2 n/256$ and $\mathbb{E}[Tot(s)] \leq 16\alpha \log^2 n$. Finally, we get:

$$\mathbb{P}(Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}) \leq \frac{16\alpha \log^2 n}{\frac{\epsilon_2 \log^2 n}{256}} = \frac{4096\alpha}{\epsilon_2}$$

□

Proof of Lemma 2. In Corollary 2.2 we have upper bounded the probability that an α-good vertex encounters many odd cycles. However, not every vertex is α-good. Therefore, we also need to consider the probability that $s \in V$ is a bad vertex. Since our goal is to upper bound, we assume that every $s \in V$ that is a bad vertex ends up seeing many odd cycles. Then:

$$\mathbb{P}(Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}) \leq \mathbb{P}(Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256} | s \in V_{good}) + \mathbb{P}(s \notin V_{good})$$

We have the upper bound of the first term from eq.13 and the second term from eq.4. When we put these together, we have that:

$$\mathbb{P}(Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}) \leq \frac{4096\alpha}{\epsilon_2} + \alpha = \frac{O(\alpha)}{\epsilon_2}$$

□

Fact. *Let $X_1, X_2, \ldots, X_{|S|}$ be independent variables in $[0, 1]$. Let $\mu := \mathbb{E}[\sum_i X_i]$.*

$$\text{For } k \geq 6\mu, \quad \mathbb{P}[X \geq k] \leq 2^{-k} \quad [1] \tag{14}$$

Proof of Theorem (Case 1). From eq.10 we have that:

$$\mathbb{P}(s \text{ is an exotic vertex}) \leq \frac{O(\alpha)}{\epsilon_2}$$

We denote the event that i-th call to `CountOddCycle` (2) resulting in an exotic vertex as X_i. And we let $X = \sum_i X_i$. In this case:

$$\mu = \mathbb{E}[\sum_i X_i] = \sum_i \mathbb{E}[X_i] \leq \sum_i \frac{O(\alpha)}{\epsilon_2} = \frac{1}{\epsilon_1} \frac{O(\alpha)}{\epsilon_2}$$

Now, we will apply the Chernoff bound from eq.14 with $k = 1/2\epsilon_1$. This selection implies that at least half of the trials need to fail for the total failure. Before proceeding, we will fix $\alpha = 1/\log n$ and $\epsilon_1 = 1/\log^4 n$. However, we need to be careful. So far, we have given three promises regarding these variables and we need to make sure that they hold.

- For the Chernoff bound to hold, we need $k \geq 6\mu$.
- For Claim 1.3 to hold, we need $0 < l\epsilon_1/\alpha^2 < 1$.
- Throughout the whole section, we promise that $0 < \alpha < 1$.

It is simple to verify that all these conditions hold with our selection of variables for sufficiently large n. We can continue with the application of the Chernoff bound:

$$\mathbb{P}(X \geq \frac{1}{2\epsilon_1}) \leq 2^{-1/2\epsilon_1} = 2^{-\log^4 n/2} \leq \frac{1}{n}$$

This implies that the graph will be rejected with probability $\leq 1/n$ (i.e. it will be accepted with probability $> 1 - 1/n > 2/3$). □

5 Case 2: $\mathcal{C}ut(G) \leq 1 - \epsilon_2$

Lemma 3. *If $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$, then:*

$$r_s \geq \frac{\epsilon_2}{64n} \quad \forall s \in V \tag{15}$$

Definition 3.1. Let $N_s^{even}(v) = V_s^{even} \cap N(v)$ where $N(v)$ denotes the neighbourhood of v.

Claim 3.1. *If $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$, $s \in V$, $v \in V_s^{even}$ and $u \in N_s^{even}(v)$, then:*

$$p_{s,v}^{even}(l) \geq \frac{1}{4n} \qquad p_{s,v}^{odd}(l) \geq \frac{|N_s^{even}(v)|}{16dn} \tag{16}$$

Proof of Claim 3.1. We first pick a starting vertex s. Then, we pick a vertex v. We assume $v \in V_s^{even}$ without loss of generality. Finally, we pick another vertex $u \in N_s^{even}(v)$. By eq.1 we have that $\forall s, v \in V \ p_{s,v}(l) \geq 1/2n$. Since $p_{s,v}^{even}(l) \geq p_{s,v}^{odd}(l)$ and $p_{s,u}^{even}(l) \geq p_{s,u}^{odd}(l)$, this implies that:

$$p_{s,v}^{even}(l) \geq \frac{1}{4n} \qquad p_{s,u}^{even}(l) \geq \frac{1}{4n}$$

We also claim that $p_{s,u}^{even}(l-1) \geq 1/8n$. This is true because from eq.1 we have that $p_{s,u}(l) \geq 1/2n$ and since $u \in V_{even}$, $p_{s,u}^{even}(l) \geq 1/4n$. Note that $p_{s,u}^{even}(l-1) \geq p_{s,u}^{even}(l)/2$ because an $l-1$ length walk can be turned into an l length walk with same parity and endpoint by adding a self-loop. The mentioned inequality holds because the probability of a self-loop is at least $1/2$. By the lazy random walk transition probabilities, we have that:

$$p_{s,v}^{odd}(l) \geq \sum_{u \in N_s^{even}(v)} \frac{1}{2d} p_{s,u}^{even}(l-1) \geq \frac{|N_s^{even}(v)|}{16dn}$$

□

Corollary 3.1. *If $G = (V, E)$ is a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$, $s \in V$, $v \in V_s^{odd}$ and $u \in N_s^{odd}(v)$, then:*

$$p_{s,v}^{odd}(l) \geq \frac{1}{4n} \qquad p_{s,v}^{even}(l) \geq \frac{|N_s^{odd}(v)|}{16dn} \tag{17}$$

Proof of Corollary 3.1. The proof follows exactly the same steps as the Proof of Claim 3.1. □

Proof of Lemma 3. First, we split the sum over two sets:

$$r_s = \sum_{v \in V} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) = \sum_{v \in V_s^{even}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) + \sum_{v \in V_s^{odd}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l)$$

We can substitute the results obtained in eq.16 and eq.17:

$$\begin{aligned}
&\sum_{v \in V_s^{even}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) + \sum_{v \in V_s^{odd}} p_{s,v}^{even}(l) p_{s,v}^{odd}(l) \\
&\geq \sum_{v \in V_s^{even}} \frac{1}{4n} \frac{|N_s^{even}(v)|}{16dn} + \sum_{v \in V_s^{odd}} \frac{1}{4n} \frac{|N_s^{odd}(v)|}{16dn} \\
&= \frac{1}{64dn^2} [\sum_{v \in V_s^{even}} |N_s^{even}(v)| + \sum_{v \in V_s^{odd}} |N_s^{odd}(v)|]
\end{aligned}$$

The quantity

$$\sum_{v \in V_s^{even}} |N_s^{even}(v)| + \sum_{v \in V_s^{odd}} |N_s^{odd}(v)|$$

is exactly the number of violating edges for the partition (V_s^{even}, V_s^{odd}). One of our starting assumptions was $\mathcal{C}ut(G) \leq 1-\epsilon_2$ which means that the number of violating edges must be at least $\epsilon_2 dn$ for all partitions. Even if the partition (V_s^{even}, V_s^{odd}) is the optimal partition:

$$\sum_{v \in V_s^{even}} |N_s^{even}(v)| + \sum_{v \in V_s^{odd}} |N_s^{odd}(v)| \geq \epsilon_2 dn$$

When we combine the results, we get that:

$$r_s \geq \frac{1}{64dn^2}\epsilon_2 dn = \frac{\epsilon_2}{64n}$$

□

Lemma 4. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$, let $s \in V$. Then the expected value that* `CountOddCycle` *(2) returns is:*

$$\mathbb{P}(Tot(s) < \frac{\epsilon_2 \log^2 n}{256}) \leq \frac{O(1)}{\epsilon_2^2 \log^2 n} \tag{18}$$

Definition 4.1 (Exotic Vertex)**.** Here, we provide the definition of exotic vertex again to remind the reader. A vertex $s \in V$ is said to be exotic with respect to `CountOddCycle` (2) if the returned value from the routine is large. Specifically, $s \in V$ is said to be exotic if:

$$Tot(s) \geq \frac{\epsilon_2 \log^2 n}{256}$$

Corollary 4.1. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$, let $s \in V$. Then:*

$$\mathbb{P}(s \textit{ is not an exotic vertex}) \leq \frac{O(1)}{\epsilon_2^2 \log^2 n} \tag{19}$$

Proof of Corollary 4.1. This result is a direct consequence of Lemma 2 □

Claim 4.1. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1-\epsilon_2$. Then during the* `CountOddCycle` *(2) routine:*

$$\mathbb{E}[\eta_{ij}] \geq \frac{\epsilon_2}{32n} \quad \forall i, j \in [t] \tag{20}$$

Proof of Claim 4.1. The probability of two different random walks creating an odd cycle is equal to two random walks ending at any same vertex with different parities:

$$\mathbb{E}[\eta_{ij}] = 2 \sum_{v \in V} p_{s,v}^{even}(l) p_{s,v}^{odd}(l)$$

The factor of two comes from the fact that either of the walks can be even. The summation is exactly equal to r_s. Therefore, using eq.15, we say that:

$$\mathbb{E}[\eta_{ij}] = 2r_s \geq \frac{\epsilon_2}{32n}$$

□

Claim 4.2. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$. Then the expected value that* `CountOddCycle` *(2) returns is:*

$$\mathbb{E}[Tot(s)] \geq \frac{\epsilon_2 \log^2 n}{128} \tag{21}$$

Proof of Claim 4.2. We first use the fact that $Tot(s)$ is equal to summation of η_{ij}'s:

$$\mathbb{E}[Tot(s)] = \mathbb{E}[\sum_{1\leq i<j\leq t} \eta_{ij}] = \sum_{1\leq i<j\leq t} \mathbb{E}[\eta_{ij}]$$

We substitute the result from eq.20 and use the fact that there are more than $t^2/4$ pairs of walks:

$$\sum_{1\leq i<j\leq t} \mathbb{E}[\eta_{ij}] \geq \frac{t^2}{4}\frac{\epsilon_2}{32n} = \frac{t^2\epsilon_2}{128n}$$

Now, we substitute the number of walks $t = \sqrt{n}\log n$:

$$\mathbb{E}[Tot(s)] \geq \frac{n\log^2 n\epsilon_2}{128n} = \frac{\epsilon_2 \log^2 n}{128}$$

□

Claim 4.3. *Let $G = (V, E)$ be a d-regular $\Omega(1)$-expander graph with $\mathcal{C}ut(G) \leq 1 - \epsilon_2$. Then the variance of the value returned by* `CountOddCycle` *(2) is:*

$$Var[Tot(s)] \leq 4\log^2 n \tag{22}$$

Proof of Claim 4.3. By the definition of variance, we have that:

$$Var[Tot(s)] = \mathbb{E}[Tot(s)^2] - \mathbb{E}[Tot(s)]^2$$

We will now deal with the two terms individually. We start with upper-bounding $\mathbb{E}[Tot(s)^2]$:

$$\begin{aligned}
\mathbb{E}[Tot(s)^2] &= \mathbb{E}[(\sum_{1\leq i<j\leq t} \eta_{ij})^2] \\
&= \sum_{1\leq i<j\leq t} \mathbb{E}[\eta_{ij}^2] + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ \{i,j\}\cap\{i,j\}=\emptyset}} \mathbb{E}[\eta_{ij}\eta_{kl}] + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ i=k}} \mathbb{E}[\eta_{ij}\eta_{kl}] \\
&= \sum_{1\leq i<j\leq t} \mathbb{E}[\eta_{ij}] + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ \{i,j\}\cap\{i,j\}=\emptyset}} \mathbb{E}[\eta_{ij}]\mathbb{E}[\eta_{kl}] + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ i=k}} \mathbb{E}[\eta_{ij}\eta_{kl}] \\
&\leq t^2\frac{2}{n} + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ \{i,j\}\cap\{i,j\}=\emptyset}} \mathbb{E}[\eta_{ij}]\mathbb{E}[\eta_{kl}] + 2t^3\frac{4}{n^2}
\end{aligned}$$

Then, we will lower-bound $\mathbb{E}[Tot(s)]^2$:

$$\begin{aligned}
\mathbb{E}[Tot(s)]^2 &= \mathbb{E}[\sum_{1\leq i<j\leq t} \eta_{ij}]^2 = (\sum_{1\leq i<j\leq t} \mathbb{E}[\eta_{ij}])^2 \\
&= \sum_{1\leq i<j\leq t} \mathbb{E}[\eta_{ij}]^2 + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ \{i,j\}\cap\{i,j\}=\emptyset}} \mathbb{E}[\eta_{ij}]\mathbb{E}[\eta_{kl}] + 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ i=k}} \mathbb{E}[\eta_{ij}]\mathbb{E}[\eta_{kl}] \\
&\geq 2\sum_{\substack{1\leq i<j\leq t\\ 1\leq k<l\leq t\\ \{i,j\}\cap\{i,j\}=\emptyset}} \mathbb{E}[\eta_{ij}]\mathbb{E}[\eta_{kl}]
\end{aligned}$$

Notice that when we subtract the terms, we can cancel out the unopened sum from the first term:

$$Var[Tot(s)] = \mathbb{E}[Tot(s)^2] - \mathbb{E}[Tot(s)]^2 \leq \frac{2t^2}{n} + \frac{8t^3}{n^2}$$

Now, we substitute the number of walks $t = \sqrt{n}\log n$:

$$Var[Tot(s)] \leq \frac{2n\log^2 n}{n} + \frac{8n^{3/2}\log^3 n}{n^2} = 2\log^2 n + \frac{8\log^3 n}{\sqrt{n}}$$

For sufficiently large n, the claim holds:

$$Var[Tot(s)] \leq 2\log^2 n + 2\log^2 n = 4\log^2 n$$

□

Proof of Lemma 4. We can apply Chebyshev's inequality on $Tot(s)$ using the results obtained in eq.21 and eq.22. Set $\mu := \mathbb{E}[Tot(s)]$ and $\sigma := \sqrt{Var[Tot(s)]}$:

$$\mathbb{P}(|Tot(s) - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

We select $k = \epsilon_2 \log n/512$. When we substitute, we get that:

$$\mathbb{P}(|Tot(s) - \frac{\epsilon_2\log^2 n}{128}| \geq \frac{\epsilon_2\log^2 n}{256}) \leq \frac{512^2}{\epsilon_2^2\log^2 n} = \frac{O(1)}{\epsilon_2^2\log^2 n}$$

This proves the lemma. □

Proof of Theorem (Case 2). From eq.19 we have that:

$$\mathbb{P}(s \text{ is not an exotic vertex}) \leq \frac{O(1)}{\epsilon_2^2\log^2 n}$$

We denote the event that i-th call to `CountOddCycle` (2) resulting in a non-exotic vertex as X_i. And we let $X = \sum_i X_i$. In this case:

$$\mu = \mathbb{E}[\sum_i X_i] = \sum_i \mathbb{E}[X_i] \leq \sum_i \frac{O(1)}{\epsilon_2^2\log^2 n} = \frac{1}{\epsilon_1}\frac{O(1)}{\epsilon_2^2\log^2 n}$$

Now, we will apply the Chernoff bound from eq.14 with $k = 1/2\epsilon_1$. This selection implies that at least half of the trials need to fail for the total failure. Before proceeding, we will fix $\epsilon_1 = 1/\log^4 n$. However, we need to be careful. We need to make sure that the following is true:

- For the Chernoff bound to hold, we need $k \geq 6\mu$.

It is simple to verify that this condition holds with our selection of variables for sufficiently large n. We can continue with the application of the Chernoff bound:

$$\mathbb{P}(X \geq \frac{1}{2\epsilon_1}) \leq 2^{-1/2\epsilon_1} = 2^{-\log^4 n/2} \leq \frac{1}{n}$$

This implies that the graph will be rejected with probability $\leq 1/n$ (i.e. it will be accepted with probability $> 1 - 1/n > 2/3$). □

6 Future Work

We have presented a sublinear-time algorithm for testing whether a bounded degree expander graph has a max-cut value close to 1 or far from 1. Even though this seems like a small subset of all graphs, we believe that it is possible to generalize from the expander case. We follow a similar analysis to the Goldreich and Ron's analysis for the expander case. Goldreich and Ron have proven that their tester works for the general case [5]. We believe that, with some care we can generalize our tester as well.

Furthermore, we realize that some of the bounds provided during the analysis may not be tight. We believe that with further care, it may be possible to decrease the logarithmic factor in $\epsilon_1 = O(1/\log^4 n)$. Ideally, we would aim for $\epsilon_1 = O(\epsilon_2^2/\log n)$

References

[1] Devdatt Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms.* Cambridge University Press, 2012.

[2] Arijit Ghosh, Gopinath Mishra, Rahul Raychaudhury, and Sayantan Sen. Tolerant bipartiteness testing in dense graphs, 04 2022.

[3] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Proceedings of 37th Conference on Foundations of Computer Science*, pages 339–348, 1996.

[4] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing - STOC '97*, 1997.

[5] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree grahps. *Combinatorica*, 19(3):335–373, 1999.

[6] Michal Parnas, Dana Ron, and Ronitt Rubenfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.