Lower Bounds for Testing Bipartiteness in Dense Graphs

Andrej Bogdanov[*] Luca Trevisan[†]

November 14, 2002

Abstract

We consider the problem of testing bipartiteness in the adjacency matrix model. The best known algorithm, due to Alon and Krivelevich, distinguishes between bipartite graphs and graphs that are ϵ-far from bipartite using $\tilde{O}(1/\epsilon^2)$ queries. We show that this is optimal for non-adaptive algorithms, up to polylogarithmic factors. We also show a lower bound of $\Omega(1/\epsilon^{3/2})$ for adaptive algorithms.

1 Introduction

The problem of *testing bipartiteness* in the *adjacency matrix* model is the problem of designing a randomized algorithm with the following properties. The algorithm is given oracle access to the adjacency matrix of an undirected graph $G = (V, E)$ and is also given a parameter $\epsilon > 0$; the algorithm is required to accept with probability at least 3/4 if the graph G is bipartite, and to reject with probability at least 3/4 if the graph G is ϵ-far from bipartite, meaning that one has to remove more than $\epsilon\binom{|V|}{2}$ edges from G in order to make it bipartite. There is no requirement on the algorithm when the given graph G is not bipartite but it can be made bipartite by removing less than $\epsilon\binom{|V|}{2}$ edges. If the algorithm accepts bipartite graphs with probability 1, then we say that it has *one-sided error.*

Goldreich, Goldwasser and Ron [GGR98] introduced this problem, as a special case of their general framework of *graph property testing*, and showed that it can be solved by a one-sided error algorithm in time polynomial in $1/\epsilon$ and independent of the size of the graph. Their algorithm simply picks a random induced subgraph with $\tilde{O}(1/\epsilon^2)$ vertices and checks whether the subgraph is bipartite. Notice that the algorithm not only has one-sided error, but is also *non-adaptive*, that is, it decides all at once which entries of the adjacency matrix to inspect.

Alon and Krivelevich [AK02] improve the result of Goldreich and el. by showing that, in fact, it is enough to look at a random subgraph with $\tilde{O}(1/\epsilon)$ vertices. Thus the algorithm looks at $\tilde{O}(1/\epsilon^2)$ entries of the adjacency matrix and runs in time $\tilde{O}(1/\epsilon^2)$. Also the algorithm, again, has one-sided error and is non-adaptive.

Alon and Krivelevich [AK02] show that there are graphs that are ϵ-far from bipartite but such that by picking $o(1/\epsilon)$ vertices at random and considering the subgraph induced by them, we see a bipartite subgraph with high probability.

[*] adib@cs.berkeley.edu. Computer Science Division, University of California, Berkeley.

[†] luca@cs.berkeley.edu. Computer Science Division, University of California, Berkeley. Work supported by NSF grant CCR 9984703 and a Sloan Research Fellowship.

Goldreich and Trevisan [GT01] prove that if there is a one-sided error algorithm for testing bipartiteness[1] having query complexity q, then it also works to just pick $2q$ vertices at random and check whether they induce a bipartite graph.

Together, these results imply that any one-sided algorithm for testing bipartiteness must have query complexity (and thus, running time) at least $\Omega(\epsilon)$. Notice that there is a quadratic gap between this lower bound and the performance of the algorithm of [AK02]. Furthermore, the case of algorithms with two-sided error is not addressed.

In this paper we show that any non-adaptive algorithm must have query complexity $\Omega(1/\epsilon^2)$ and any algorithm must have query complexity $\Omega(1/\epsilon^{1.5})$.

Our "hard instances" for this problem are random graphs where every edge exists with probability $2\epsilon + o(1)$. With high probability, such graphs are ϵ-far from being bipartite.

Consider now the simplest case, that of a *one-sided error non-adaptive algorithm*, and let us see what happens for a fixed randomness of the algorithm and over the choices of the random graph. The algorithm looks at q pairs of vertices, and each pair is going to be connected by an edge with probability 2ϵ. Basically, the view of the algorithm is a fixed graph with q edges, from which each edge is being deleted independently with probability $1-2\epsilon$, and kept with probability 2ϵ. We are able to argue that if we start from an arbitrary graph with $q = o(1/\epsilon^2)$ edges, then, after the deletions, the graph is very likely to become a forest, and, therefore, to be bipartite.

Regarding *one-sided error adaptive* algorithms, consider again the view of the algorithm for a fixed randomness of the algorithm and a random graph. Every time the algorithm makes a query into the adjacency matrix, it discovers an edge with probability 2ϵ and it finds out that there is no edge with probability $1 - 2\epsilon$. When the algorithm discovers a cycle, it is because it makes a query (u, v) where u and v were already discovered to be connected, and then (u, v) turns out to be an edge. Typically, the algorithm will have to make $\Omega(1/\epsilon)$ such attempts before discovering a cycle. So, by the time a cycle is discovered, the algorithm must have found enough edges that there are $\Omega(1/\epsilon)$ pairs of connected vertices. Then, the algorithm must have discovered at least $\Omega(1/\sqrt{\epsilon})$ edges to account for so much connections, and therefore it must have made $\Omega(1/\epsilon^{1.5})$ queries into the adjacency matrix. We can conclude that an algorithm that makes $o(1/\epsilon^{1.5})$ queries is very likely to see a forest, and, therefore, a bipartite graph.

The analysis of algorithms with *two-sided error* is more involved. We need to consider two distributions of graphs, one made of bipartite graphs and one made of graphs that are typically ϵ-far from bipartite, and then argue that the distributions are indistinguishable for algorithms of small query complexity.

We take again the random graph with edge probability $2\epsilon + o(1)$ as one distribution (this one will typically contain graphs ϵ-far from bipartite). The other distribution is sampled as follows: we first randomly partition the vertices of the graphs, and then each edge crossing the partition is picked independently with probability $4\epsilon + o(1)$, and all other edges are not picked (by construction, these graphs are bipartite).

Roughly speaking, we show that conditioned on the event of seeing a forest, the views of an algorithm when given oracle access to a graph chosen from one distribution or from the other have statistical distance $o(1)$, and we already know that the condition holds with probability $1 - o(1)$ for adaptive algorithms making $o(1/\epsilon^{1.5})$ queries and non-adaptive

[1] Their result holds for any graph property testing problem in this model, but for simplicity we state here only the application to bipartiteness.

algorithms making $o(1/\epsilon^2)$ queries.[2]

We note that there is an adaptive algorithm that discovers odd cycles in time $O(1/\epsilon^{1.5})$ when given a random graph with edge probability 2ϵ. In fact, for any distribution that we could think of to produce graphs that are ϵ-far from bipartite, there is always an adaptive algorithm that discovers odd cycles in time $O(1/\epsilon^{1.5})$. It would be very interesting to come up with an adaptive algorithm of query complexity $o(1/\epsilon^2)$: Goldreich and Trevisan show that there is always at most a quadratic gap between the complexity of adaptive versus non-adaptive algorithms, but there is no natural example that we know of where an actual gap occurs.[3]

We know of no previous paper proving a lower bound super-linear in $1/\epsilon$ for a property that can be tested in time polynomial in $1/\epsilon$, and of no super-linear bound for two-sided error algorithms for any property testable in time depending only on ϵ. Alon [Alo01] proves a super-polynomial lower bound in $1/\epsilon$ for one-sided error algorithms for a class of problems that admit algorithms running in time that depends only on ϵ. Alon's lower bound, however, does not apply to two-sided error algorithms.

We also note that our results imply that the problem of testing whether a graph is a forest has query complexity $\Omega(1/\epsilon^{1.5})$ for adaptive one-sided error algorithms and $\Omega(1/\epsilon^2)$ for non-adaptive one-sided error algorithms, while it is trivially testable in time $O(1/\epsilon)$ by a non-adaptive two-sided error algorithm. This gives a separation between the power of one-sided versus two-sided error algorithms for a natural problem. Regarding one-sided error algorithms, it is easy to come up with a $O((1/\epsilon^2)\log 1/\epsilon)$ non-adaptive algorithm for finding a cycle in a graph that is ϵ-far from being a forest. It would be interesting to come up with a $o(1/\epsilon^2)$ adaptive one-sided error algorithm, that would show a separation between the power of adaptive versus non-adaptive algorithms. Perhaps this is an easier question to attack than the design of a $o(1/\epsilon^2)$ adaptive algorithm for bipartiteness.

2 Lower bounds for testers with one-sided error

In this section we show that any property testing algorithm for bipartiteness with one-sided error must perform $\Omega(1/\epsilon^{3/2})$ queries. Moreover, if the algorithm is nonadaptive, then it must perform $\Omega(1/\epsilon^2)$ queries.

Let A be a one-sided property testing algorithm for bipartiteness that performs q queries. Let $q_i = (x_i, y_i)$ denote the i-th query of A, and $Q = \{q_i : 1 \leq i \leq q\}$ denote the set of all queries of A. Without loss of generality, we may assume that all queries q_i are distinct, hence $|Q| = q$. We observe that if A rejects an input G, then A must have detected a witness that refutes the bipartiteness of G; it follows that $E(G) \cap Q$ contains an odd cycle. Therefore, to show a one-sided lower bound of q queries for testing bipartiteness, it is enough to exhibit a distribution $\mathcal{G}$ on n-vertex graphs such that: (1) With probability $1 - o(1)$, graph $G \sim \mathcal{G}$ is ϵ-far from bipartite, and (2) With probability $2/3$, the set $E(G) \cap Q$ contains no odd cycle.

[2] This is just a simplified account of the proof. The result would hold as stated, and in fact the distance in the conditional distributions would be zero, if the "view" of the algorithm were just the set of edges it discovers. But, in addition, the algorithm also discovers that certain pairs of vertices are not connected. To account for that we need some additional conditioning, and even then we can show that the distance is $o(1)$ but not zero.

[3] Here, of course, we are referring to the case of graph property testing in the *adjacency matrix* model. It is easy to come up with huge gaps in the *adjacency list* model of [GR97].

Let $\mathcal{G}$ denote the distribution on n-vertex graphs where each edge is selected independently at random with probability $p = 2\epsilon + O(\sqrt{\epsilon/n})$. Using a standard probabilistic argument, it is not difficult to show that with probability $1 - o(1)$, a graph chosen from $\mathcal{G}$ is ϵ-far from bipartite. We now argue that $E(G) \cap Q$ is unlikely contain an odd cycle, whenever $q = \Omega(1/\epsilon^{3/2})$ for adaptive algorithms and $q = \Omega(1/\epsilon^2)$ for nonadaptive algorithms.

Theorem 1. *Let A be an adaptive property testing algorithm, and Q denote the set of queries of A on input $G \sim \mathcal{G}$, where $|Q| = q \leq 1/24\epsilon^{3/2}$. With probability $2/3$ over the choice of G and the randomness of A, the graph $G' = (V(G), E(G) \cap Q)$ is a forest.*

Proof. Let $Q_t = \{q_1, \dots, q_t\}$, and $G_t = (V(G), E(G) \cap Q_t)$. We call query $q_t = (x_t, y_t)$ *internal* if x_t and y_t belong to the same connected component of G_{t-1}. Note that the first query that reveals a cycle in G', if such a query exists, must be internal. We will show that, with probability $5/6$, the number of distinct internal queries in Q is less than $1/7\epsilon$. Since distinct queries are satisfied independently with probability p, it follows that all the internal queries in Q are negative with probability at least $(1 - 1/7\epsilon)^p > 5/6$. Therefore the probability that G' contains no cycle is at least $\frac{5}{6} \cdot \frac{5}{6} > \frac{2}{3}$.

We now bound the number q_I of distinct internal queries in Q. Let $s_1, \dots, s_k$ denote the sizes (number of edges) of the connected components of G' that contain at least one edge. The number of pairs of vertices in the same component of G' that are not connected by an edge in G' is at most $S = \sum_{i=1}^{k} \binom{s_i}{2}$. If query $q_t = (x_t, y_t)$ is internal, then x_t and y_t belong to the same connected component of G'. It follows that $q_I \leq S$.

By linearity of expectation, $\mathrm{E}[s_1 + \ldots + s_k] = p(q - q_I) \leq pq$, as each non-internal query contributes to this sum with probability p. Therefore $s_1 + \ldots + s_k \leq 6qp$ with probability at least $5/6$. Finally, with probability $5/6$,

$$S \leq \binom{s_1 + \ldots + s_k}{2} \leq \binom{6qp}{2} \leq 1/7\epsilon. \quad \square$$

We observe that the bound in the theorem is tight for the distribution $\mathcal{G}$ up to a constant factor. In fact, there is an adaptive algorithm that finds a triangle in G with probability $2/3$ and $O(1/\epsilon^{3/2})$ queries. The algorithm consists of two phases: In the first phase, the algorithm makes $q_1 = O(1/\epsilon^{3/2})$ queries of type (v, v_i), where $v, v_1, \dots, v_{q_1}$ are arbitrary distinct vertices. With probability $5/6$, at least $O(1/\sqrt{\epsilon})$ of the answers are positive. In the second phase, the algorithm makes $q_2 = O(1/\epsilon)$ distinct queries among pairs (v_i, v_j) such that $(v, v_i), (v, v_j) \in E(G)$. With probability $5/6$, this phase reveals an edge (v_i, v_j), and therefore a triangle v, v_i, v_j.

Theorem 2. *Let A be a nonadaptive property testing algorithm, and Q denote the set of queries of A on input $G \sim \mathcal{G}$, where $|Q| = q \leq 1/73\epsilon^2$. With probability $2/3$ over the choice of G, the graph $G' = (V(G), E(G) \cap Q)$ is a forest.*

To prove the theorem, we will make use of the following combinatorial result:

Lemma 1. *A graph with m edges has fewer than $(2m)^{l/2}$ cycles of length l.*

Proof. Let A be the adjacency matrix of a graph with m edges, and let $\lambda_1, \dots, \lambda_n$ denote the eigenvalues of A. We note that the number of ordered cycles of length l is the trace of A^l. By a standard inequality,

$$\text{trace}A^l = \sum_{i=1}^{n} \lambda_i^l \leq \left(\sum_{i=1}^{n} \lambda_i^2 \right)^{l/2} = (\text{trace}A^2)^{l/2} = (2m)^{l/2}. \quad \square$$

Proof of Theorem 2. We bound the expected number of cycles of length l in G'. By Lemma 1, the set of queries determines at most $(2q)^{l/2}$ cycles of length l; each cycle has probability p^l to be present in G. Therefore the expected number of cycles of length l in G' is at most $(2q)^{l/2}p^l < 1/3^l$ for large enough n.

It follows that the expected number of cycles of any length in G' is at most $\sum_{l=3}^{\infty} 1/3^l < 1/3$. By Markov's inequality, G' has no cycles with probability at least $2/3$. □

3 Lower bounds for testers with two-sided error

In this section we extend the lower bounds from Theorems 1 and 2 to algorithms for testing bipartiteness that may exhibit two-sided error. To prove a two-sided lower bound of q for testing bipartiteness, we need to argue that a sequence of q queries cannot distinguish bipartite graphs from graphs that are ϵ-far from bipartite with statistical significance better than, say, 1/3. A one-sided tester is more restricted than a two-sided tester in the sense that it must find evidence of bipartiteness in the form of an odd cycle. The information obtained from negative answers to its queries is not significant in this context. For two-sided testers, however, absence of evidence is not evidence of absence.[4] A two-sided error tester may take advantage of the negative queries to infer statistical properties of its input.

The *transcript* $\text{tr}_A(G)$ of algorithm A on input G consists of a sequence of queries $\mathbf{q} = (q_i : 1 \leq i \leq q)$ and answers $\mathbf{a} = (a_i : 1 \leq i \leq q)$, where $a_i = 1$ if $q_i \in G$, and $a_i = 0$ otherwise. In adaptive algorithms, the query q_i may depend on previous queries $(q_1, \ldots, q_{i-1})$ and answers $(a_1, \ldots, a_{i-1})$. To prove that there is no q query tester for bipartiteness with success probability δ it is sufficient, by Yao's principle, to produce two distributions of graph instances $\mathcal{G}$ and $\mathcal{H}$ with the following properties:

1. With high probability, a graph selected from $\mathcal{G}$ is ϵ-far from bipartite.

2. A graph selected from $\mathcal{H}$ is bipartite.

3. For any deterministic algorithm A, the statistical distance between transcripts of A on input $G \sim \mathcal{G}$ and $G \sim \mathcal{H}$ is at most δ:

$$\frac{1}{2} \sum_{\mathbf{q},\mathbf{a}} | \Pr_{G \sim \mathcal{G}}[\text{tr}_A(G) = (\mathbf{q}, \mathbf{a})] - \Pr_{G \sim \mathcal{H}}[\text{tr}_A(G) = (\mathbf{q}, \mathbf{a})]| \leq \delta.$$

We will use the following two distributions of instances: $\mathcal{G}$ is the distribution defined in Section 2, namely random graphs on n vertices with edge probability $p = 2\epsilon + O(\sqrt{\epsilon/n})$. We define $\mathcal{H}$ to be the distribution of random *bipartite* graphs with edge probability $2p$. More precisely, a graph $G \sim \mathcal{H}$ is generated as follows: (1) Pick a partition $(S, \bar{S})$ of $V(G)$ uniformly at random; (2) Select each edge $(u \in S, v \in \bar{S})$ independently at random with probability $2p$.

[4] Secretary of defense D. Rumsfeld, on possible links between Al Qaeda and Iraq.

Say G is *consistent* with transcript $(\mathbf{q}, \mathbf{a})$ if $a_i = 1$ when q_i is an edge of G, and $a_i = 0$ otherwise. It is not difficult to estimate the probability that a graph $G \sim \mathcal{G}$ is consistent with $\mathbf{q}, \mathbf{a}$; this probability depends only on the number of positive answers a_i. When $G \sim \mathcal{H}$, however, the answers a_i are not independent. For example, the event "There is a path of length 2 between u and v" biases the probability of an edge between u and v. Even though the structure of $\mathcal{H}$ makes direct computations difficult, we single out a class of *typical* transcripts which have approximately the same probability in both distributions. We then argue that a testing algorithm with suitably low query complexity is likely to produce a typical transcript. To simplify notation, we ignore constants in our analysis.

For a transcript $(\mathbf{q}, \mathbf{a})$, we partition the queries in $\mathbf{q}$ as follows:

1. The set of *positive queries* Q^+ is the set of queries q_i such that $a_i = 1$.

2. The set of *internal negative queries* Q_I^- is the set of queries $q_i = (u_i, v_i)$ such that $a_i = 0$ and u_i, v_i are connected by a path in Q^+.

3. The set of *external negative queries* Q_E^- is the set of queries $q_i = (u_i, v_i)$ such that $a_i = 0$ and u_i, v_i are not connected by a path in Q^+.

Let $q^+ = |Q^+|$, $q_I^- = |Q_I^-|$ and $q_E^- = |Q_E^-|$. Let $\mathcal{C}$ denote the class of connected components of $G^+ = (V(G), Q^+)$. For $U, V \in \mathcal{C}$, let $e_{UV} = |\{(u, v) \in Q_E^- : u \in U, v \in V\}|$. We call transcript $(\mathbf{q}, \mathbf{a})$ *typical* if: (1) G^+ is a forest, (2) $q_I^- = o(1/\epsilon)$ and (3) $\sum_{U,V \in \mathcal{C}} e_{UV}^2 = o(1/\epsilon^2)$.

The following lemma shows that if algorithm A produces a typical transcript of length $o(1/\epsilon^2)$, then it cannot determine the distribution of its input.

Lemma 2. *For any algorithm A and typical transcript $(\mathbf{q}, \mathbf{a})$ of length $q = o(1/\epsilon^2)$, $\Pr_{G \sim \mathcal{G}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})] \sim \Pr_{G \sim \mathcal{H}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})]$.*

Proof. If $G \sim \mathcal{G}$, its edges are selected independently, so G is consistent with $\mathbf{q}, \mathbf{a}$ with probability $p^{q^+}(1-p)^{q_I^- + q_E^-}$. For $G \sim \mathcal{H}$, we write $\Pr_{G \sim \mathcal{H}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})]$ as a product of three terms:

$$\Pr_{\mathcal{H}}[Q^+ \subseteq E(G)] \Pr_{\mathcal{H}}[Q_I^- \subseteq E(\bar{G}) | Q^+ \subseteq E(G)] \Pr_{\mathcal{H}}[Q_E^- \subseteq E(\bar{G}) | Q^+ \subseteq E(G) \cap Q_I^- \subseteq E(\bar{G})].$$

We estimate each of these probabilities. Since Q^+ is a forest, $\Pr_{\mathcal{H}}[Q^+ \subseteq E(G)] = p^{q^+}$. The second probability is a product of q_I^- terms of value either 1 (for odd length paths) or $1-2\epsilon$ (for even length paths). Since $q^+ = o(1/\epsilon)$, $\Pr_{\mathcal{H}}[Q_I^- \subseteq E(\bar{G}) | Q^+ \subseteq E(G)] \sim (1-p)^{q_I^-} \sim 1$.

For the last probability, consider a random partition $(S, \bar{S})$ of $V(G)$ that is consistent with Q^+. For every pair $U, V \in \mathcal{C}$, let E_{UV} be the number of edges in Q_E^- between U and V that are partitioned by $(S, \bar{S})$. With respect to the choice of partition, the E_{UV} are pairwise independent and

$$\mathrm{Var}[\sum_{U,V} E_{UV}] = \sum_{U,V} \mathrm{Var}[E_{UV}] \leq \sum_{U,V} e_{UV}^2 = o(1/\epsilon^2).$$

By Chebyshev's inequality, almost surely $|\sum_{U,V} E_{UV} - q_E^-/2| = o(1/\epsilon)$. In other words, for almost every partition $(S, \bar{S})$, roughly half of the edges in Q_E^- fall across the partition and roughly half fall within the partition. Therefore,

$$\Pr_{\mathcal{H}}[Q_E^- \subseteq E(\bar{G}) | Q^+ \subseteq E(G) \cap Q_I^- \subseteq E(\bar{G})] \sim (1-2p)^{q_E^-/2 \pm o(1/\epsilon)} \sim (1-p)^{q_E^-}.$$

It follows that the G is consistent with $\mathbf{q}, \mathbf{a}$ with asymptotically identical probabilities according to $\mathcal{G}$ and $\mathcal{H}$. □

The next two lemmas justify the use of the word "typical" to describe transcripts of A. They show that if A has suitably low query complexity, then it is likely to produce a typical transcript.

Lemma 3. *For any adaptive algorithm A with query complexity $q \leq o(1/\epsilon^{3/2})$ and a graph $G \sim \mathcal{G}$, the transcript $\mathrm{tr}_A(G)$ is typical with probability $1 - o(1)$.*

Proof. Let G^+ denote the subgraph of G whose vertices are endpoints of queries in $\mathbf{q}$ and with edges Q^+. Let s_i denote the size of the ith connected component of G^+ containing at least one edge, and $S = s_1 + \ldots, s_k$. By linearity of expectation, with respect to either distribution, $\mathrm{E}[(s_1 - 1) + \ldots + (s_k - 1)] \leq pq$, so that almost surely $S = O(qp) = o(1/\sqrt{\epsilon})$. As in the proof of Theorem 1, the number of internal queries is $o(1/\epsilon)$, so that almost surely all internal queries fail and G^+ is a forest. On the other hand, the number of negative internal queries cannot exceed $o(1/\epsilon)$. This establishes properties (1) and (2) of typical transcripts.

For property (3), let $\sum_{U,V \in \mathcal{C}} e_{UV}^2 = S_1 + S_2 + S_3$, where $S_1 = \sum_{|U|=|V|=1} e_{UV}^2$, $S_2 = \sum_{|U|,|V| \geq 2} e_{UV}^2$ and $S_3 = \sum_{|U|=1,|V|\geq 2} e_{UV}^2$. We bound each of the sums S_1, S_2 and S_3. In S_1, each of the terms is 0 or 1, and the number of terms is at most q. Therefore $S_1 \leq q = o(1/\epsilon^{3/2})$. For S_2 we note that

$$\sum_{|U|,|V|\geq 2} e_{UV}^2 \leq \left(\sum_{|U|,|V|\geq 2} e_{UV} \right)^2 \leq \binom{S}{2}^2 = o(1/\epsilon^2).$$

To compute S_3, we let $e_U = \sum_{|V|\geq 2} e_{UV}$. Then $e_U \leq S$, and

$$\sum_{|U|=1,|V|\geq 2} e_{UV}^2 \leq \sum_{|U|=1} e_U^2 \leq S \sum_{|U|=1} e_U \leq S \cdot q = o(1/\epsilon^2). \quad \square$$

Lemma 4. *For any nonadaptive algorithm A with query complexity $q \leq o(1/\epsilon^2)$ and a graph $G \sim \mathcal{H}$, the transcript $\mathrm{tr}_A(G)$ is typical with probability $1 - o(1)$.*

Proof. Let $G' = (V(G), Q^+ \cup Q_I^- \cup Q_E^-)$. Property (1) is proved as in Theorem 2. To show (2), for every $e \in Q_I^-$, let X_{el} denote the number of paths of length l between the endpoints of e. Every X_{el} is a sum of indicator random variables Y_c, one for each cycle c of length $l+1$ that contains e, such that $Y_c = 1$ if all edges in c except possibly e are in Q^+. It follows that $\Pr[Y_c = 1] = p^l$.

$$\begin{aligned}
\mathrm{E}[\sum_{l=1}^{\infty} \sum_{e \in Q_I^-} X_{el}] &\leq \sum_{l=1}^{\infty} \sum_{e \in E(G')} \sum_{c \ni e, |c| = l+1} \Pr[Y_c = 1] \\
&= \sum_{l=1}^{\infty} \sum_{c, |c|=l+1} l p^l \\
&\leq \sum_{l=1}^{\infty} l (2q)^{(l+1)/2} p^l \text{ (by Lemma 1)} \\
&= o(1/\epsilon).
\end{aligned}$$

By Markov's inequality, almost always $q_I^- = o(1/\epsilon)$.

For property (3), given any pair $e = (u, v), e' = (u', v') \in Q_E^-$, let $X_{ee'l}$ denote the number of pairs of paths (u, v) and (u', v') whose lengths sum to l. For a pair of components $U, V \in \mathcal{C}$,

$$e_{UV}^2 \leq \sum_{l=1}^{\infty} \sum_{u,u' \in U, v, v' \in V} X_{(u,v)(u',v')l}.$$

Again, we write $X_{ee'l}$ as a sum of indicator random variables Y_c, one for each cycle c of length $l + 2$ that contains both e and e'. For a fixed c, there are at most $l(l-1)$ pairs (e, e') such that Y_c is an indicator for $X_{ee'l}$.

$$\begin{aligned}
\mathrm{E}[\sum_{l=1}^{\infty} \sum_{U,V \in \mathcal{C}} \sum_{u,u \in U, v, v' \in V} X_{(u,v)(u',v')l}] &\leq \mathrm{E}[\sum_{l=1}^{\infty} \sum_{e,e' \in E(G')} X_{ee'l}] \\
&\leq \sum_{l=1}^{\infty} \sum_{e,e' \in E(G')} \sum_{c \ni e, e', |c| = l+2} \Pr[Y_c = 1] \\
&\leq \sum_{l=1}^{\infty} \sum_{c, |c| = l+2} l(l-1)p^l \\
&\leq \sum_{l=1}^{\infty} l(l-1)(2q)^{(l+2)/2} p^l \\
&= o(1/\epsilon^2).
\end{aligned}$$

By Markov's inequality, property (3) holds almost always. □

Theorem 3. *Any algorithm A for testing bipartitenes has query complexity $\Omega(1/\epsilon^{3/2})$. Moreover, if A is nonadaptive, then it has query complexity $\Omega(1/\epsilon^2)$.*

Proof. Suppose that A is an adaptive tester for bipartiteness with query complexity $o(1/\epsilon^{3/2})$, and G be an input. By Lemma 3, if $G \sim \mathcal{G}$, the transcript of A on G is almost surely typical. This is also the case if $G \sim \mathcal{H}$:

$$\begin{aligned}
\Pr_{G \sim \mathcal{H}}[\mathrm{tr}_A(G) \text{ is typical}] &= \sum_{(\mathbf{q}, \mathbf{a}) \text{ typical}} \Pr_{G \sim \mathcal{H}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})] \\
&\sim \sum_{(\mathbf{q}, \mathbf{a}) \text{ typical}} \Pr_{G \sim \mathcal{G}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})] \text{ (by Lemma 2)} \\
&= \Pr_{G \sim \mathcal{G}}[\mathrm{tr}_A(G) \text{ is typical}] \\
&= 1 - o(1).
\end{aligned}$$

By Lemma 2, a typical transcript is asymptotically equiprobable for $G \sim \mathcal{G}$ and $G \sim \mathcal{H}$, and it follows that

$$\frac{1}{2} \sum_{\mathbf{q}, \mathbf{a}} | \Pr_{G \sim \mathcal{G}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})] - \Pr_{G \sim \mathcal{H}}[\mathrm{tr}_A(G) = (\mathbf{q}, \mathbf{a})]| = o(1).$$

The analysis for nonadaptive testers is identical. □

References

[AK02] N. Alon and M. Krivelevich. Testing k-colorability. *SIAM J. on Discrete Mathematics*, 15:211–227, 2002.

[Alo01] Noga Alon. Testing subgraphs in large graphs. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 434–441, 2001.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connections to learning and approximation. *Journal of the ACM*, 45:653–750, 1998.

[GR97] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 289–298, 1997.

[GT01] Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 460–469, 2001.