Algorithmic Aspects of Property Testing in the Dense Graphs Model

Oded Goldreich*
Department of Computer Science
Weizmann Institute of Science
Rehovot, ISRAEL.
oded.goldreich@weizmann.ac.il

Dana Ron†
Department of EE–Systems
Tel-Aviv University
Ramat-Aviv, ISRAEL.
danar@eng.tau.ac.il

January 18, 2010

Abstract

In this paper we consider two basic questions regarding the query complexity of testing graph properties in the adjacency matrix model. The first question refers to the relation between adaptive and non-adaptive testers, whereas the second question refers to testability within complexity that is inversely proportional to the proximity parameter, denoted ϵ. The study of these questions reveals the importance of algorithmic design in this model. The highlights of our study are:

- A gap between the complexity of adaptive and non-adaptive testers. Specifically, there exists a natural graph property that can be tested using $\widetilde{O}(\epsilon^{-1})$ adaptive queries, but cannot be tested using $o(\epsilon^{-3/2})$ non-adaptive queries.
- In contrast, there exist natural graph properties that can be tested using $\widetilde{O}(\epsilon^{-1})$ non-adaptive queries, whereas $\Omega(\epsilon^{-1})$ queries are required even in the adaptive case.

We mention that the properties used in the foregoing conflicting results have a similar flavor, although they are of course different.

This article is an extended abstract of our technical report [GR08]. While the main text assume familiarity with the basic model, all relevant definitions appear in Section 5.1.

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a couple of recent surveys [R1, R2]). Loosely speaking, property testing typically refers to sub-linear time probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain bits of the object by performing queries, which means that the object is seen as a function and the testers get oracle access to this function. Thus, a tester may be expected to work in time that is sub-linear in the length of the description of this object.

Much of the aforementioned work (see, e.g., [GGR, AFKS, AFNS]) was devoted to the study of testing graph properties in the adjacency matrix model, which is also the setting of the current work. In this model, introduced in [GGR], graphs are viewed as symmetric Boolean functions

*Partially supported by the Israel Science Foundation (grants No. 460/05 and 1041/08).
†Partially supported by the Israel Science Foundation (grants No. 89/05 and 246/08).

over a domain consisting of all possible vertex-pairs. Namely, an N-vertex graph $G = ([N], E)$ is represented by the function $g : [N] \times [N] \to \{0, 1\}$ such that $\{u, v\} \in E$ if and only if $g(u, v) = 1$. Consequently, an N-vertex graph represented by the function $g : [N] \times [N] \to \{0, 1\}$ is said to be ϵ-far from some predetermined graph property if more than $\epsilon \cdot N^2$ entries of g must be modified in order to yield a representation of a graph that has this property. We refer to ϵ as the proximity parameter, and the complexity of testing is stated in terms of ϵ and the number, N, of vertices in the graph.

Interestingly, many natural graph properties can be tested within query complexity that depends only on the proximity parameter; see [GGR], which presents testers with query complexity $\mathrm{poly}(1/\epsilon)$, and [AFNS], which characterizes the class of properties that are testable within query complexity that depends only on the proximity parameter (where this dependence may be an arbitrary function of ϵ). However, a common phenomenon in all the aforementioned works is that they utilize quite naive algorithms and their focus is on the analysis of these algorithms, which is often quite sophisticated. This phenomenon is no coincidence: As shown in [AFKS, GT], when ignoring a quadratic blow-up in the query complexity, property testing in this model reduces to sheer combinatorics. Specifically, without loss of generality, the tester may just inspect a random induced subgraph (of adequate size) of the input graph.

In this paper we demonstrate that a more refined study of property testing in this model reveals the importance of algorithmic design also in this model. This is demonstrated both by studying the advantage of adaptive testers over non-adaptive ones as well as by studying the class of properties that can be tested within complexity that is inversely proportional to the proximity parameter.

2 Two Related Studies

We start by reviewing the two related studies conducted in the current work.

2.1 Adaptivity vs. Non-adaptivity

A tester is called non-adaptive if it determines all its queries independently of the answers obtained for previous queries, and otherwise it is called adaptive. Indeed, by [AFKS, GT], the benefit of adaptivity (or, equivalently, the cost of non-adaptivity) is polynomially bounded: Specifically, any (possibly adaptive) tester, for any graph property, of query complexity $q(N, \epsilon)$ can be transformed into a non-adaptive tester of query complexity $O(q(N, \epsilon)^2)$. But is this quadratic gap an artifact of the known proofs (of [AFKS, GT]) or does it reflect something inherent?

A recent work by [GR07] suggests that the latter case may hold: For every $\epsilon > 0$, they showed that the set of N-vertex bipartite graphs of maximum degree $O(\epsilon N)$ is ϵ-testable (i.e., testable with respect to proximity parameter ϵ) by $\widetilde{O}(\epsilon^{-3/2})$ queries, while by [BT] a non-adaptive tester for this set must use $\Omega(\epsilon^{-2})$ queries. Thus, there exists a case where non-adaptivity has the cost of increasing the query complexity; specifically, for any $c < 4/3$, the query complexity of the non-adaptive tester is greater than a c-power of the query complexity of the adaptive tester (i.e., $\widetilde{O}(\epsilon^{-3/2})^c = o(\epsilon^{-2})$). We stress that the result of [GR07] does not refer to property testing in the "proper" sense; that is, the complexity is not analyzed with respect to a varying value of the proximity parameter for a fixed property. It is rather the case that, for every value of the proximity parameter, a different property, which depends on this parameter, is considered. The upper bounds and lower bounds refer to this combination of a property tailored for a fixed value of the proximity parameter. Thus, *the work of [GR07] leaves open the question of whether there exists a single graph property such that adaptivity is beneficial for any value of the proximity parameter* (as long as $\epsilon > N^{-\Omega(1)}$). That is, the question is whether adaptivity is beneficial for the standard asymptotic-complexity formulation of property testing.

2.2 Complexity linearly related to the proximity parameter

As shown in [GGR], many natural graph properties can be tested within query complexity that is polynomial in the reciprocal of the proximity parameter and independent of the size of the graph. We ask whether a linear complexity is possible at all, and if so which properties can be tested with query complexity that is linear (or almost linear) in the reciprocal of the proximity parameter, that is, with query complexity $\tilde{O}(1/\epsilon)$.[1]

The first question is easy to answer even when avoiding *trivial* properties. We say that a graph property Π is trivial for testing if for every $\epsilon > 0$ there exists $N_0 > 0$ such that for every $N \geq N_0$ either all N-vertex graphs belong to Π or all of them are ϵ-far from Π. Note that the property of being a clique (equiv., an independent set) can be tested by $O(1/\epsilon)$ queries, even when these queries are non-adaptive (e.g., make $O(1/\epsilon)$ random queries and accept if and only if all return 1). Still, we ask whether "more interesting" graph theoretical properties can also be tested within similar complexity, either only adaptively or also non-adaptively. In particular, the property of being a clique (or an independent set) is viewed as "non-interesting" since it contains a single N-vertex graph (per each N) and is represented by a monochromatic function.

3 Our Results

We address the foregoing questions by studying a sequence of natural graph properties, which are defined formally in Section 5.2. The first property in the sequence, called clique collection and denoted $\mathcal{CC}$, is the set of graphs such that each graph consists of a collection of isolated cliques. Testing this property corresponds to the following natural clustering problem: can a set of possibly related elements be partitioned into "perfect clusters" (i.e., two elements are in the same cluster if and only if they are related)? For this property, $\mathcal{CC}$, we prove a gap between adaptive and non-adaptive query complexity, where the adaptive query complexity is almost linear in the reciprocal of the proximity parameter. That is:

Theorem 3.1 (the query complexity of clique collection):

1. *There exists an adaptive tester of query complexity $\widetilde{O}(\epsilon^{-1})$ for $\mathcal{CC}$. Furthermore, this tester has one-sided error and runs in time $\widetilde{O}(\epsilon^{-1})$.*[2]
2. *Any non-adaptive tester for $\mathcal{CC}$ must have query complexity $\Omega(\epsilon^{-4/3})$.*
3. *There exists a non-adaptive tester of query complexity $O(\epsilon^{-4/3})$ for $\mathcal{CC}$. Furthermore, this tester has one-sided error and runs in time $O(\epsilon^{-4/3})$.*

Note that the complexity gap between Parts 1 and 2 of Theorem 3.1 matches the gap established by [GR07] for "non-proper" testing. A larger gap is established for a property of graphs, called bi-clique collection and denoted $\mathcal{BCC}$, where a graph is in $\mathcal{BCC}$ if it consists of a collection of isolated bi-cliques (i.e., complete bipartite graphs). We note that bi-cliques may be viewed as the bipartite analogues of cliques (w.r.t. general graphs), and indeed they arise naturally in clustering applications that are modeled by bipartite graphs over two types of elements.

[1] Note that $\Omega(1/\epsilon)$ queries are required for testing any of the graph properties considered in the current work.

[2] We refer to a model in which elementary operations regarding pairs of vertices are charged at unit cost.

Theorem 3.2 (the query complexity of bi-clique collection):

1. *There exists an adaptive tester of query complexity $\widetilde{O}(\epsilon^{-1})$ for $\mathcal{BCC}$. Furthermore, this tester has one-sided error and runs in time $\widetilde{O}(\epsilon^{-1})$.*

2. *Any non-adaptive tester for $\mathcal{BCC}$ must have query complexity $\Omega(\epsilon^{-3/2})$. Furthermore, this holds even if the input graph is promised to be bipartite.*

The furthermore clause in Part 2 of Theorem 3.2 holds also for the model studied in [AFN], where the bi-partition of the graph is given.

Theorem 3.2 asserts that the gap between the query complexity of adaptive and non-adaptive testers may be a power of $1.5 - o(1)$. Recall that the results of [AFKS, GT] assert that the gap may not be larger than quadratic. We conjecture that this upper bound can be matched.

Conjecture 3.3 (an almost-quadratic complexity gap): *For every positive integer $t \geq 5$, there exists a graph property Π for which the following holds:*

1. *There exists an adaptive tester of query complexity $\widetilde{O}(\epsilon^{-1})$ for Π.*

2. *Any non-adaptive tester for Π must have query complexity $\Omega(\epsilon^{-2+(2/t)})$.*

3. *There exists an efficient non-adaptive tester of query complexity $\widetilde{O}(\epsilon^{-2+2t^{-1}})$ for Π.*

Furthermore, Π consists of graphs that are each a collection of "super-cycles" of length t, where a super-cycle is a set of t independent sets arranged on a cycle such that each pair of adjacent independent sets is connected by a complete bipartite graph.

We were able to prove Part 2 of Conjecture 3.3, but failed to provide a full analysis of an algorithm that we designed for Part 1. However, we were able to prove a *promise problem version of Conjecture 3.3*; specifically, this promise problem (stated in Theorem 5.4) refers to inputs promised to reside in a set $\Pi' \supset \Pi$ and the tester is required to distinguish graphs in Π from graphs that are ϵ-far from Π.

In contrast to the foregoing results that aim at identifying properties with a substantial gap between the query complexity of adaptive versus non-adaptive testing, we also study cases in which no such gap exists. Since query complexity that is linear in the reciprocal of the proximity parameter is minimal for many natural properties, and, in fact, for any property that is "non-trivial for testing" (as defined at the end of Section 2), we focus on non-adaptive testers that approximately meet this bound. Among the results obtained in this direction, we highlight the following one.

Theorem 3.4 (the query complexity of collections of $O(1)$ cliques): *For every positive integer c, there exists a non-adaptive tester of query complexity $\widetilde{O}(\epsilon^{-1})$ for the set of graphs such that each graph consists of a collection of up to c cliques. Furthermore, this tester has one-sided error and runs in time $\widetilde{O}(\epsilon^{-1})$.*

Theorem 3.4 should be viewed as a first step in the study of graph properties that are the simplest to test; that is, the class of graph properties that have a non-adaptive of query complexity $\widetilde{O}(\epsilon^{-1})$. We mention that a second step, which significantly generlaizes Theorem 3.4, has been subsequently taken in [A09, AG].

Discussion. The foregoing results demonstrate that a finer look at property testing of graphs in the adjacency matrix model reveals the role of algorithm design in this model. In particular, in some cases (see, e.g., Theorems 3.1 and 3.2), carefully designed adaptive algorithms outperform any non-adaptive algorithm. Indeed, this conclusion stands in contrast to [GT, Thm. 2], which suggests that a less fine view, which ignores polynomial blow-ups,[3] deems algorithm design irrelevant to this model. We also note that, in some cases (see, e.g., Theorem 3.4 and Part 3 of Theorem 3.1), carefully designed non-adaptive algorithms outperform canonical ones.

As discussed previously, one of the goals of this work was to study the relation between adaptive and non-adaptive testers in the adjacency matrix model. Our results demonstrate that, in this model, the relation between the adaptive and non-adaptive query-complexities is not fixed, but rather varies with the computational problem at hand. In some cases (e.g., Theorem 3.4) the complexities are essentially equal, indeed, as in the case of sampling [CEG]. In other cases (e.g., Theorem 3.1), these complexities are related by a fixed power (e.g., $4/3$) that is strictly between 1 and 2. And, yet, in other cases (e.g., Theorem 5.4) the non-adaptive complexity is quadratic in the adaptive complexity, which is the maximum gap possible (by [AFKS, GT]). Furthermore, by Theorem 5.4, for any $t \geq 4$, there exists a promise problem for which the aforementioned complexities are related by a power of $2-(2/t)$.

Needless to say, the fundamental relation between adaptive and non-adaptive algorithms was studied in a variety of models, and the current work studies it in a specific natural model (i.e., of property testing in the adjacency matrix representation). In particular, this relation has been studied in the context of property testing in other domains. Specifically, in the setting of testing the satisfiability of linear constraints, it was shown that adaptivity offers absolutely no gain [BHR]. A similar result holds for testing monotonicity of sequences of positive integers [F04]. In contrast, an exponential gap between the adaptive and non-adaptive complexities may exist in the context of testing other properties of functions [F04]. Lastly, we mention that an even more dramatic gap exists in the setting of testing graph properties in the bounded-degree model (of [GR02]); see [RS06].

4 A complexity theoretic perspective

Let us start by rephrasing Conjecture 3.3, while recalling that it refers to properties for which testing requires (adaptive) query complexity that is at least linear in the reciprocal of the proximity parameter (see Proposition 5.2).

Conjecture 3.3 (rephrased). *For every integer $t \geq 2$, there exists a* (natural) *graph property Π_t such that* non-adaptively *testing Π_t has query complexity $\widetilde{\Theta}(q^{2-(2/t)})$, where $q = q(N, \epsilon)$ denotes the the query complexity of* (adaptively) *testing Π_t.*

Recall that it is known that the non-adaptive query complexity of testing any graph property is at most quadratic in the adaptive query complexity. We stress that Conjecture 3.3 not only asserts that this upper bound is essentially tight, but rather asserts an infinite hierarchy of possible functional relations between the non-adaptive and adaptive query complexity.

The results in this work refer to "two and a half" elements in the conjectured hierarchy as well as to a corresponding hierarchy of promise problems. Specifically, denoting the (adaptive) query complexity by $q = q(N, \epsilon)$, we have:

[3] Recall that [GT, Thm. 2] asserts that canonical testers, which merely select a random subset of vertices and rule according to the induced subgraph, have query-complexity that is at most quadratic in the query-complexity of the best tester. We note that [GT, Thm. 2] also ignores the time-complexity of the testers.

- Theorem 3.4 establishes the conjecture for $t = 2$. Specifically, Theorem 3.4 presents natural graph properties that have non-adaptive query complexity $\widetilde{\Theta}(q)$.
- Theorem 3.1 establishes the conjecture for $t = 3$. Specifically, Theorem 3.1 presents a natural graph property that has non-adaptive query complexity $\widetilde{\Theta}(q^{4/3})$.
- Theorem 3.2 establishes half of the conjecture for $t = 4$. Specifically, Theorem 3.2 presents a natural graph property that has non-adaptive query complexity $\widetilde{\Omega}(q^{3/2})$.
- Theorem 5.4 fully establishes the conjecture in the setting of promise problems. We stress that these promise problems are fixed (independently of the proximity parameter).

Indeed, in all our results $q = q(N, e) = \widetilde{\Omega}(1/\epsilon)$. We also mention that in all our results the upper bounds are established by one-sided error testers, whereas the lower bounds hold also for general (i.e., two-sided error) testers.

Open problems. In addition to the resolution of Conjecture 3.3, our study raises many other open problems; the most evident ones are listed next.

1. What is the non-adaptive query complexity of $\mathcal{BCC}$? Note that Theorem 3.2 only establishes a lower bound of $\Omega(\epsilon^{-3/2})$. We conjecture that an efficient non-adaptive algorithm of query complexity $\widetilde{O}(\epsilon^{-3/2})$ can be devised.
2. For which constants $c \in [1, 2]$ does there exist a property that has adaptive query complexity of $q(\epsilon)$ and non-adaptive query complexity of $\widetilde{\Theta}(q(\epsilon)^c)$? Note that Theorem 3.1 shows that $4/3$ is such a constant, and the same holds for the constant 1 (see, e.g., Theorem 3.4). We conjecture (see Conjecture 3.3) that, for any $t \geq 2$, it holds that the constant $2 - (2/t)$ also satisfies the foregoing requirement. It may be the case that these constants are the only ones that satisfy this requirement.
3. Characterize the class of graph properties for which the query complexity of non-adaptive testers is almost linear in the query complexity of adaptive testers.
4. Characterize the class of graph properties for which the query complexity of non-adaptive testers is almost quadratic in the query complexity of adaptive testers.
5. Characterize the class of graph properties for which the query complexity of adaptive (resp., non-adaptive) testers is almost linear in the reciprocal of the proximity parameter.

The last characterization project may be the most feasible among the three foregoing characterization projects. We mention that this is partially addressed in [A09, AG], which significatly extends and build upon Theorem 3.4. Finally, we recall the well-known open problem, partially addressed in [AS], of providing a characterization of the class of graph properties that are testable within query complexity that is polynomial in the reciprocal of the proximity parameter.

5 Appendix

In this section we review the definition of property testing, when specialized to graph properties in the adjacency matrix model. We also define several natural graph properties, which serve as the pivot of our study, and state some additional results.

5.1 Basic notions

For an integer n, we let $[n] = \{1, \ldots, n\}$. A generic N-vertex graph is denoted by $G = ([N], E)$, where $E \subseteq \{\{u, v\} : u, v \in [N]\}$ is a set of unordered pairs of vertices. Any set of such graphs that is closed under isomorphism is called a graph property. By oracle access to such a graph $G = ([N], E)$ we mean oracle access to the Boolean function that answers the query $\{u, v\}$ (or rather $(u, v) \in [N] \times [N]$) with the bit 1 if and only if $\{u, v\} \in E$.

Definition 5.1 (property testing for graphs in the adjacency matrix model): *A* tester *for a graph property* Π *is a probabilistic oracle machine that, on input parameters* N *and* ϵ *and access to an* N*-vertex graph* $G = ([N], E)$*, outputs a binary verdict that satisfies the following two conditions.*

1. *If* $G \in \Pi$ *then the tester accepts with probability at least* $2/3$.
2. *If* G *is* ϵ*-far from* Π *then the tester accepts with probability at most* $1/3$*, where* G *is* ϵ-far *from* Π *if for every* N*-vertex graph* $G' = ([N], E') \in \Pi$ *it holds that the symmetric difference between* E *and* E' *has cardinality that is greater than* ϵN^2.[4]

If the tester accepts every graph in Π *with probability 1, then we say that it has* one-sided error. *A tester is called* non-adaptive *if it determines all its queries based solely on its internal coin tosses* (and the parameters N and ϵ)*; otherwise it is called* adaptive.

The query complexity of a tester is the number of queries it makes to any N-vertex graph oracle, as a function of the parameters N and ϵ. We say that a tester is efficient if it runs in time that is polynomial in its query complexity, where basic operations on elements of $[N]$ (and in particular, uniformly selecting an element in $[N]$) are counted at unit cost. We note that all testers presented in this paper are efficient, whereas the lower bounds hold also for non-efficient testers.

We shall focus on properties that can be tested with query complexity that only depends on the proximity parameter, ϵ. Thus, the query complexity upper bounds that we state hold for any values of ϵ and N, but will be meaningful only for $\epsilon > 1/N^2$ or so. In contrast, the lower bounds (e.g., of $\Omega(1/\epsilon)$) cannot possibly hold for $\epsilon < 1/N^2$, but they will indeed hold for any $\epsilon > N^{-\Omega(1)}$. Alternatively, one may consider the query-complexity as a function of ϵ, where for each fixed value of $\epsilon > 0$ the value of N tends to infinity.

Notation and a convention. For a fixed graph $G = ([N], E)$, we denote by $\Gamma(v) = \{u : \{u, v\} \in E\}$ the set of neighbors of vertex v. At times, we look at E as a subset of $V \times V$; that is, we often identify E with $\{(u, v) : \{u, v\} \in E\}$. If a graph $G = ([N], E)$ is not ϵ-far from a property Π then we say that G is ϵ-close to Π; this means that at most ϵN^2 edges should be added and/or removed from G such to yield a graph in Π.

5.2 The graph properties to be studied

The set of graphs that consists of a collection of isolated cliques is called clique collection and is denoted $\mathcal{CC}$; that is, a graph $G = ([N], E)$ is in $\mathcal{CC}$ if and only if the vertex set $[N]$ can be partitioned into $(C_1, \ldots, C_t)$ such that the subgraph induced by each C_i is a clique and there are no edges with endpoints in different C_i's (i.e., for every $u < v \in [N]$ it holds that $\{u, v\} \in E$ if and only if there

[4]Indeed, it is more natural to require that this symmetric difference should have cardinality that is greater than $\epsilon \cdot \binom{N}{2}$. The current convention is adopted for the sake of convenience.

exists an i such that $u, v \in C_i$). If $t \leq c$ then we say that G is in $\mathcal{CC}^{\leq c}$; that is, $\mathcal{CC}^{\leq c}$ is the subset of $\mathcal{CC}$ that contains graphs that are each a collection of up-to c isolated cliques.

A bi-clique is a complete bipartite graph (i.e., a graph $G = (V, E)$ such that V is partitioned into $(S, V \setminus S)$ such that $\{u, v\} \in E$ if and only if $u \in S$ and $v \in V \setminus S$). Note that a graph is a bi-clique if and only if its complement is in $\mathcal{CC}^{\leq 2}$. The set of graphs that consists of a collection of isolated bi-cliques is called bi-clique collection and denoted $\mathcal{BCC}$; that is, a graph $G = ([N], E)$ is in $\mathcal{BCC}$ if and only if the vertex set $[N]$ can be partitioned into $(V_1, \ldots, V_t)$ such that the subgraph induced by each V_i is a bi-clique and there are no edges with endpoints in different V_i's (i.e., each V_i is partitioned into $(S_i, V_i \setminus S_i)$ such that for every $u < v \in [N]$ it holds that $\{u, v\} \in E$ if and only if there exists an i such that $(u, v) \in S_i \times (V_i \setminus S_i)$).

Generalizations of $\mathcal{BCC}$ are obtained by considering collections of "super-paths" and "super-cycles" respectively. A super-path (of length t) is a sequence of disjoint sets of vertices, $S_1, \ldots, S_t$, such that vertices $u, v \in \bigcup_{i \in [t]} S_i$ are connected by an edge if and only if for some $i \in [t-1]$ it holds that $u \in S_i$ and $v \in S_{i+1}$. Note that a bi-clique can be viewed as a super-path of length two. We denote the set of graphs that consists of a collection of isolated super-paths of length t by $\mathcal{S}\mathsf{P}_t\mathcal{C}$ (e.g., $\mathcal{S}\mathsf{P}_2\mathcal{C} = \mathcal{BCC}$). Similarly, a super-cycle (of length t) is a sequence of disjoint sets of vertices, $S_1, \ldots, S_t$, such that vertices $u, v \in \bigcup_{i \in [t]} S_i$ are connected by an edge if and only if for some $i \in [t]$ it holds that $u \in S_i$ and $v \in S_{(i \bmod t)+1}$. Note that a bi-clique that has at least two vertices on each side can be viewed as a super-cycle of length four (by partitioning each of its sides into two parts). We denote the set of graphs that consists of a collection of isolated super-cycles of length t by $\mathcal{S}\mathsf{C}_t\mathcal{C}$ (e.g., $\mathcal{S}\mathsf{C}_4\mathcal{C} \subset \mathcal{BCC}$, where the strict containment is due to the pathological case of bi-cliques having at most one node on one side).

5.3 Additional results

In this section we state two siomple lower bounds as well as the promise problem version of Conjecture 3.3.

Lower bounds. We first note that $\Omega(1/\epsilon)$ (adaptive) queries are required for testing any graph property that is non-trivial for testing, where a graph property Π is non-trivial for testing if there exists $\epsilon_0 > 0$ such that for infinitely many $N \in \mathbb{N}$ there exist N-vertex graphs G_1 and G_2 such that $G_1 \in \Pi$ and G_2 is ϵ_0-far from Π. We note that all properties considered in this work are non-trivial for testing. On the other hand, the negation of this (non-triviality) condition means that for every $\epsilon > 0$ and all sufficiently large $N \in \mathbb{N}$ either Π contains no N-vertex graph or all N-vertex graphs are ϵ-close to Π. In such a case (for every such ϵ and N), the tester may decide without even looking at the graph.[5] Turning back to properties that are non-trivial for testing, we prove that any tester for such a property must have query complexity $\Omega(1/\epsilon)$.

Proposition 5.2 *Let Π be a property that is non-trivial for testing. Then, any tester for Π has query complexity $\Omega(1/\epsilon)$.*

Note that the claim holds also for general properties (i.e., arbitrary sets of functions). To justify the fact that all our testers are inherently non-canonical, we show that (for any property that is non-trivial for testing) canonical testers must use $\Omega(\epsilon^{-2})$ queries.

Proposition 5.3 *Let Π be a property that is non-trivial for testing. Then, any canonical tester for Π has query complexity $\Omega(1/\epsilon^2)$.*

[5]Indeed, there exists natural graph properties that are trivial for testing (e.g., connectivity, non-planarity, having no vertex of odd degree); see [GGR, Sec. 10.2.1].

The promise problem version of Conjecture 3.3. For every positive integer $t \geq 4$, we consider a promise problem, denoted Π_t, having inputs that are either in $\mathcal{S}\mathtt{C}_t\mathcal{C}$ or in some specific subset, denoted $\mathcal{S}\mathtt{C}_{2t}\mathcal{C}'$, of $\mathcal{S}\mathtt{C}_{2t}\mathcal{C}$. On proximity parameter ϵ, a tester of Π_t is required to accept inputs in $\mathcal{S}\mathtt{C}_t\mathcal{C}$ and reject inputs in $\mathcal{S}\mathtt{C}_{2t}\mathcal{C}'$ that are ϵ-far from $\mathcal{S}\mathtt{C}_t\mathcal{C}$.

Theorem 5.4 (an almost-quadratic complexity gap for promise problems): *For every positive integer $t \geq 4$, the promise problem Π_t satisfies the following:*

1. *There exists an adaptive tester of query complexity $\widetilde{O}(\epsilon^{-1})$ for Π_t. Furthermore, this tester runs in time $\widetilde{O}(\epsilon^{-1})$.*
2. *Any non-adaptive tester for Π_t must have query complexity $\Omega(\epsilon^{-2+(2/t)})$.*
3. *There exists a non-adaptive tester of query complexity $O(\epsilon^{-2+(2/t)})$ for Π_t. Furthermore, this tester runs in time $O(\epsilon^{-2+(2/t)})$.*

References

[A81] N. Alon. On the number of subgraphs of prescribed type of graphs with a given number of edges. *Israel J. Math. 38*, pages 116–130, 1981.

[AFKS] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy. Efficient Testing of Large Graphs. *Combinatorica*, Vol. 20, pages 451–476, 2000.

[AFN] N. Alon, E. Fischer, and I. Newman. Testing of bipartite graph properties. *SIAM Journal on Computing*, Vol. 37, pages 959–976, 2007.

[AFNS] N. Alon, E. Fischer, I. Newman, and A. Shapira. A Combinatorial Characterization of the Testable Graph Properties: It's All About Regularity. In *38th STOC*, pages 251–260, 2006.

[AS] N. Alon and A. Shapira. A Characterization of Easily Testable Induced Subgraphs. *Combinatorics Probability and Computing*, 15:791–805, 2006.

[A09] L. Avigad. On the Lowest Level of Query Complexity in Testing Graph Properties. Master Thesis, Weizmann Institute of Scienc, December 2009.

[AG] L. Avigad and O. Goldreich. Testing Graph Blow-Up. In preparations.

[BHR] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. 3CNF properties are hard to test. *SIAM Journal on Computing*, Vol. 35 (1), pages 1–21, 2005.

[BT] A. Bogdanov and L. Trevisan. Lower Bounds for Testing Bipartiteness in Dense Graphs. In *IEEE Conference on Computational Complexity*, pages 75–81, 2004.

[CEG] R. Canetti, G. Even and O. Goldreich. Lower Bounds for Sampling Algorithms for Estimating the Average. *IPL*, Vol. 53, pages 17–25, 1995.

[F04] E. Fischer. On the strength of comparisons in property testing. *Inform. and Comput.*, Vol. 189 (1), pages 107-116, 2004.

[GGR] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998.

[GR02] O. Goldreich and D. Ron. Property Testing in Bounded Degree Graphs. *Algorithmica*, Vol. 32 (2), pages 302–343, 2002.

[GR08] O. Goldreich and D. Ron. Algorithmic Aspects of Property Testing in the Dense Graphs Model. *ECCC*, TR08-039, 2008.

[GR09] O. Goldreich and D. Ron. On Proximity Oblivious Testing. Extended abstract in the proceedings of the *41st STOC*, 2009.

[GT] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, Vol. 23 (1), pages 23–57, August 2003.

[GR07] M. Gonen and D. Ron. On the Benefit of Adaptivity in Property Testing of Dense Graphs. In *Proc. of RANDOM'07*, LNCS Vol. 4627, pages 525–539, 2007. To appear in *Algorithmica* (special issue of RANDOM and APPROX 2007).

[R1] D. Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, Vol. 1 (3), pages 307–402, 2008.

[R2] D. Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in TCS*, to appear.

[RS06] S. Raskhodnikova and A. Smith. A note on adaptivity in testing properties of bounded-degree graphs. *ECCC*, TR06-089, 2006.

[RS96] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2), pages 252–271, 1996.