Electronic Colloquium on Computational Complexity, Report No. 88 (2021)

Open Problems in Property Testing of Graphs

Oded Goldreich[*]

June 23, 2021

Abstract

We briefly discuss a few open problems in the study of various models of testing graph properties, focusing on the query complexity of the various tasks. In the dense graph model, we discuss several open problems, including:

- Determining the complexity of testing triangle-freeness.
- Characterizing the class of properties that are testable within extremely low complexity.

Turning to the bounded-degree graph model, we discuss several open problems, including:

- Characterizing the class of properties that are testable within size-oblivious complexity.
- Determining the complexity of graph isomorphism.

In each of the foregoing models, we also discuss a favorite open problem that was recently resolved. Lastly, we discuss the vast lack of knowledge with respect to testing graph properties in the general graph model.

Contents

[*]Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. Email: oded.goldreich@weizmann.ac.il. This research was partially supported by the Israel Science Foundation (grant No. 671/13) and by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 819702).

ISSN 1433-8092

Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a recent textbook [13]). 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 local views of the object by making adequate queries; that is, the object is modeled as a function and the testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the size of the object).

A significant portion of the foregoing research has been devoted to testing graph properties in three different models: the dense graph model (introduced in [19] and reviewed in [13, Chap. 8]), the bounded-degree graph model (introduced in [21] and reviewed in [13, Chap. 9]), and the general graph model (introduced in [34, 29] and reviewed in [13, Chap. 10]). These three models have received different levels of attention, and are commonly believed to have reached different levels of understanding; nevertheless, interesting open problem abound in each of them. In this memo, we describe a few of these open problems, with an unavoidable and obvious biased towards our own work (which reflect our own directions of interest).

Each of the three foregoing models is defined by specifying the *type of queries* that testers may make (to the tested graph) and a *measure of distance* between graphs (with respect to which the performance of the testers is evaluated). For example, in the dense graph model, queries refer to the existence of edges between specified pairs of vertices (i.e., adjacency queries), and the distance between graphs refers to the fraction of vertex-pairs on which the two graphs differ. Each of these three models will be defined at the beginning of the section devoted to its review.

Two versions of each model. Like most results in the area (esp., when allowing two-sided error (see Appendix A)), the open problems that we review are valid both for the "standard version" of the foregoing three models as well as (for its restriction) to the natural model of "size-oblivious testers" (to be defined below). Both versions, which are described below, are natural special cases of a general framework presented in [15], which we briefly recall next.

Our starting point is the observation that, in all three models, it is commonly postulated that the vertex-set consists of $\{1, 2, ..., n\}$, where n is a natural number that is given explicitly to the tester, whereas in general one may be interested in corresponding graphs over any vertex-set. The foregoing simplified assumption is made in almost all studies of these models; yet, as observed in [15], this loses no generality, *provided that (1) the tester can sample the vertex-set, and (2) the tester is explicitly given the size of the vertex-set.*[1]

Having explicitly stated the two foregoing conditions that allow to extend testers of the simplified model to more general settings, we observe that these conditions are of fundamentally different nature. The first condition (i.e., sampleability of the vertex-set) seems essential to testing any non-trivial property, whereas the second condition (i.e., knowledge of the (exact) size of the vertex-set) may be relaxed and even avoided altogether in many cases.

For example, all graph-partition properties (see [19]) and all subgraph-free properties are testable in a general version of the dense graph model in which only the first condition holds. This is the case since the original testers (presented in [19] and [3], resp.) merely use the description of the vertex-set only in order to sample it. Needless to say, it follows that the query

[1]See Observations 1.2, 2.2, and 3.2 in [15].

complexities of these (non-adaptive)[2] testers are oblivious of the size of the graph (and depend only on the proximity parameter), but (as observed by [5]) the converse does not hold (i.e., testers of size-oblivious query complexity may depend on the size of the graph for their verdict (see also [27])).

On the other hand, when the query complexity depends on the size of the graph, the tester needs to obtain at least a sufficiently good approximation of the said size. Typically, such an approximation suffices, as in the case of the bipartite tester for the bounded-degree and general graph models [22, 29]. Note, however, that an approximation of the size of the graph can be obtained, with high probability, by sampling; this yields (adaptive) testers of complexity that is at least a square root of the number of vertices.

In light of the above, we highlight two cases regarding the (*a priori*) knowledge of the size of the vertex-set (where in all cases the tester is given access to samples drawn from the vertex-set):

1. The tester is explicitly given the exact size of the vertex-set.

   As shown in [15, Obs. 1.2, 2.2, and 2.3], this ("exact size") case is essentially reducible to the simplified case in which the vertex-set equals $\{1, 2, ..., n\}$ and n is explicitly given to the tester.

   Hence, this version will be called the standard version.

2. The tester is not given explicitly any information regarding the size of the vertex-set.

   Such testers will be called size-oblivious, and the corresponding version (of the testing model) will be called the size-oblivious version.

We stress that, unlike in [16], we confine ourselves here to the case that the sampling device provides uniformly and independently distributed vertices in the tested graph.[3]

On characterizations. When seeking a characterization of the class of graph properties having a specified complexity, we seek a criteria that will make it significantly easier (than the definition of the class) to determine whether a property belongs to the class. On the other hand, we are willing to restrict the class of graph properties to natural subclasses that are defined in general terms (e.g., properties in $\mathcal{P}$ or $\mathcal{NP}$, properties that admit tester of significantly higher complexity, etc). Likewise, it may be easier to provide characterization for the size-oblivious versions of the models.

On natural problems. We consider an object to be *natural with respect to a specific setting* if it was considered before in a significantly different setting. Specifically, a graph property is considered natural in the context of demonstrating the non-emptiness of a complexity class if it was considered before in studies that do not refer to that class. For example, all graph properties that are explicitly mentioned in this memo (e.g., triangle-freeness, graph isomorphism) are natural.

[2]A tester is called non-adaptive if it determines its queries based solely on its input parameters and internal randomness, but independently of answers obtained to prior queries; in other words, a non-adaptive tester determines all its queries (*a priori*) before making any query.

[3]In contrast, in [16], we considered more general models in which the tester obtains samples that are arbitrarily distributed in the vertex-set. In this case, the definition of the distance between graphs is modified to reflect this vertex distribution; such a modification is not required in the current memo.

Sticking to the basics. We avoid the temptation of considering ramifications and variants of the basic model of property testing. Specific variants that we do not discuss include tolerant testing and distance approximation [35], proximity-oblivious testers [24], and sample-based testers [25]. Each of these variants gives rise to many interesting open problems.

In contrast to the foregoing decision to narrow our scope, we include in this memo brief reviews of two open problems that were resolved recently. See paragraphs at the end of Sections 1 and 2.

Standard conventions. By graphs we mean simple (labeled) graphs; that is, self-loops and multiple edges are not allowed. A graph property is a set of graphs that is closed under isomorphism; that is, if the graph G is in the graph property Π, then so is every graph that is isomorphic to G. Throughout this memo, the proximity parameter is always denoted ϵ.

Organization. The following three sections present open problems of the three aforementioned models of testing graph properties. In Section 1 we consider the dense graph model (a.k.a. the adjacency matrix model), in Section 2 we consider the bounded-degree graph model (a.k.a. the bounded incidence lists model), and in Section 3 we consider the general graph model. The definitional parts of these sections contain some repetitions in order to enable reading them independently of one another. *We stress that all open problems refer both to the standard and to the size-oblivious versions of the models.* In Appendix A, we show that some known results that seem to rely on providing the tester with the size of the graph can actually be emulated by size-oblivious testers.

1 Testing Graph Properties in the Dense Graph Model

Here we consider the standard and the size-oblivious versions of the notion of property testing in the dense graph model (a.k.a. the adjacency matrix model, which was introduced in [19] and is reviewed in [13, Chap. 8]).

In this model, a graph of the form $G = (V, E)$ is represented by its adjacency predicate $g : V \times V \to \{0,1\}$; that is, $g(u,v) = 1$ if and only if u and v are adjacent in G (i.e., $\{u,v\} \in E$). Distance between graphs (over the same vertex-set) is measured in terms of their foregoing representation; that is, as the fraction of (the number of) entries on which they disagree (over $|V|^2$). The tester is given oracle access to the representation of the input graph (i.e., to the adjacency predicate g) as well as to a device that returns uniformly distributed elements in the graph's vertex-set. As usual, the tester is also given the proximity parameter ϵ, which determined when graphs are considered "far apart" (i.e., see the notion of ϵ-far in Definition 1.1).

In addition, in the standard version, the tester is also given the size of the vertex-set as auxiliary input. Hence, the explicit inputs given to it are $|V|$ and ϵ. In the size-oblivious version, the tester only gets ϵ as an explicit input.

Definition 1.1 (property testing in the dense graph model): *A* tester for the graph property Π (in the standard version of the dense graph model) *is a probabilistic oracle machine T that, on input a* proximity parameter $\epsilon > 0$ *and* size parameter $|V|$*, and access to two oracles, an adjacency predicate $g : V \times V \to \{0,1\}$ and a device denoted* `Samp`(V) *that samples uniformly in V, satisfies the following two conditions:*

1. *The tester accepts each graph $G = (V, E)$ in Π with probability at least $2/3$; that is, for every $g : V \times V \to \{0,1\}$ representing a graph in Π* (and every $\epsilon > 0$), *it holds that* $\Pr[T^{g,\mathtt{Samp}(V)}(|V|, \epsilon) = 1] \geq 2/3$.

2. *Given $\epsilon > 0$ and oracle access to any graph G that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $g : V \times V \to \{0,1\}$ that represents a graph that is ϵ-far from Π, it holds that $\Pr[T^{g,\mathtt{Samp}(V)}(|V|, \epsilon) = 0] \geq 2/3$, where the graph represented by $g : V \times V \to \{0,1\}$ is* ϵ-far from Π *if for every $g' : V \times V \to \{0,1\}$ that represents a graph in Π it holds that $|\{(u,v) \in V^2 : g(u,v) \neq g'(u,v)\}| > \epsilon \cdot |V|^2$.*

The tester is said to have one-sided error probability *if it always accepts graphs in Π; that is, for every $g : V \times V \to \{0,1\}$ representing a graph in Π* (and $\epsilon > 0$), *it holds that* $\Pr[T^{g,\mathtt{Samp}(V)}(|V|, \epsilon) = 1] = 1$. *The* size-oblivious version *is obtained by not providing the tester with $|V|$; that is, a* size-oblivious tester *is only given the proximity parameter $\epsilon > 0$* (and access to the oracles $g : V \times V \to \{0,1\}$ and $\mathtt{Samp}(V)$)*.*

The query complexity of a tester for Π is a function (of the parameters n and ϵ) that represents the number of queries made by the tester on the worst-case n-vertex graph, when given the proximity parameter ϵ. Our focus in this section will be on cases in which the query complexity only depends on the proximity parameter (i.e., size-oblivious query complexity), but other cases will also be considered at the end.

Well known open problems. The most famous open problem in the area is determining the query complexity of testing triangle-freeness (and other subgraph-freeness problems). It is known that this testing problem has query complexity that only depends on ϵ (see [3]), but the gap between the known lower and upper bounds is striking. On the one hand, Alon has proved that the complexity must grow faster than any polynomial in $1/\epsilon$; specifically, the lower bound is $\exp(\Omega(\log(1/\epsilon))^2)$ [2]. On the other hand, Fox proved an upper bound that is a tower of $O(\log(1/\epsilon))$ exponents [12].

Open Problem 1.2 (the complexity of testing triangle-freeness): *What is the query complexity of testing the set of graphs that contain no triangles.*

Indeed, the same question arises for testing H-freeness (i.e., having no subgraph that is isomorphic to H), for any non-bipartite graph H. (In contrast, the case of bipartite graphs H is well-understood [13, Exer. 8.13].) The complexity of testing induced subgraph freeness is also wide opened.

Another notoriously hard challenge is characterizing the class of graph properties that can be tested in query complexity $\mathrm{poly}(1/\epsilon)$.

Open Problem 1.3 (the class of properties of $\mathrm{poly}(1/\epsilon)$ query complexity): *Provide a characterization of the class of properties that can be tested using* $\mathrm{poly}(1/\epsilon)$ *queries.*

Of special interest (and possibly easier to analyze) are the special cases of query complexity $\widetilde{O}(1/\epsilon^2)$ and $\widetilde{O}(1/\epsilon)$. Turning to an even vaguer challenge, it would be nice to provide an intuitive explanation as to why H-freeness (for non-bipartite H) is harder to test than 3-Coloring (and any of the generalized partition problems); see [3, 2] versus [19].

A more refined look at the complexity of testing. A tester is called non-adaptive if it determines its queries based solely on its input parameters and internal randomness, but independently of answers obtained to prior queries; in other words, a non-adaptive tester determines all its queries (*a priori*) before making any query. Otherwise, the tester is called adaptive. For a fixed graph property Π, we denote by $q = q_\Pi$ the general (i.e., adaptive) query complexity of testing Π, and by $Q = Q_\Pi$ the corresponding non-adaptive query complexity. Recall that, for every Π, it holds that $Q = O(q^2)$ (see [3, 26]). Now, while Problem 1.3 seems a tall call, it may be more feasible to provide characterizations of much more modest classes. For example.

Open Problem 1.4 (the class of properties of extremely low complexity): *Provide a characterization of the class of properties that can be tested by non-adaptive testers of $\widetilde{O}(1/\epsilon)$ query complexity.*

We mention that this class is known to contain all sets of graphs that are obtained by (possibly unbalanced) blow-ups of a fixed graph [6].

One-sided vs two-sided error testing. One source of the gap between the complexity of one-sided and two-sided error testing is the mere formulation of the property: Properties that have an explicit quantitative nature such as the set of graphs having a clique (resp., a cut) of density at least ρ exhibit a significant gap between their one-sided and two-sided error query complexity, whereas such gaps do not exist with respect to analogous properties that have a more qualitative nature such as 3-Colorability [19]. More interesting cases of such complexity gaps refer to properties that are defined in qualitative terms such as H-freeness, for any fixed bipartite graph H. In that specific case, testing is possible with $O(1/\epsilon)$ queries (and two-sided error) by using an approximation of the edge-density in the tested graph [2] (see also [13, Exer. 8.13.2]), but one-sided error testing typically requires significantly higher query complexity [13, Exer. 8.13.3]. Intuitively, in this case, the property is approximated by an easier-to-test property, and a two-sided error tester may capitalize on that fact (while a one-sided error tester can not do so). We wonder if something more systematic and appealing can be said about this subject.

Other levels of query complexity. We mention that, although known hierarchy theorems establish the existence of properties with quite arbitrary query complexity (for testing) [20, 14], identifying natural properties of different query complexity levels is an interesting challenge. Of particular interest are complexities of the type $Q_1(n, \epsilon) = \text{poly}(\epsilon^{-1} \log n)$, $Q_2(n, \epsilon) = \exp(1/\epsilon)$, and $Q_3(n, \epsilon) = O(n^c \cdot F(1/\epsilon))$ for arbitrary $c \in (0, 2)$ and $F : \mathbb{N} \to \mathbb{N}$. Indeed, a notable example is provided by [11] who semi-determined the query complexity of (two versions of the problem) of testing graph isomorphism; specifically, ignoring factors of ϵ, they showed that:

1. The query complexity of testing isomorphism to a fixed n-vertex graph is $\widetilde{\Theta}(n^{1/2})$.[4]

2. The query complexity of testing isomorphism between two n-vertex graphs is between $\widetilde{\Omega}(n)$ and $O(n^{1+o(1)})$.[5]

In addition, for one-sided error testing, tight bounds were shown (i.e., $\widetilde{\Theta}(n)$ and $\widetilde{\Theta}(n^{3/2})$, respectively).

[4]Indeed, the fixed graph version is a "massively parametrized" property [32].

[5]The upper bound is due to [33], improving over the $\widetilde{O}(n^{5/4})$ bound of [11].

A recently resolved open problem. Recall that, for every Π, it holds that $Q = O(q^2)$, where $q = q_\Pi$ denotes the general (i.e., adaptive) query complexity of testing Π, and $Q = Q_\Pi$ denotes the corresponding *non-adaptive* query complexity. A recent work [28] proves that this bound is almost tight, in general; that is, there exists Π such that $Q = \widetilde{\Omega}(q^2)$. Furthermore, for every $c \in [1, 2]$, there exists Π such that $Q = \Theta(q^c)$ up to factors that are polylogarithmic in the size of the graph and polynomial in the proximity parameter. We wonder whether such results can be obtain without the latter slackness and for more natural graph properties. We mention that partial results of this type were reported previously in [23] (e.g., for $q = \widetilde{O}(1/\epsilon)$ and $c = 4/3$).

2 Testing Graph Properties in the Bounded-Degree Graph Model

Here we consider the standard and the size-oblivious versions of the notion of property testing in the bounded-degree graph model (a.k.a. the bounded incidence lists model, which was introduced in [21] and is reviewed in [13, Chap. 9]).

The bounded-degree graph model refers to a fixed (constant) degree bound, denoted $d \geq 2$. In this model, a graph $G = (V, E)$ of maximum degree d is represented by the incidence function $g : V \times [d] \to V \cup \{\perp\}$ such that $g(v, j) = u \in V$ if u is the j^{th} neighbor of v and $g(v, j) = \perp \notin V$ if v has less than j neighbors.[6] Distance between graphs is measured in terms of their foregoing representation; that is, as the fraction of (the number of) different array entries (over $d|V|$).

As in the dense graph model, the tester is given oracle access to the representation of the input graph (i.e., to the incidence function g) as well as to a device that returns uniformly distributed elements in the graph's vertex-set. As usual, the tester is also given the proximity parameter ϵ. In addition, in the standard version, the tester is also given the size of the vertex-set as auxiliary input. Hence, the explicit inputs given to it are $|V|$ and ϵ. In the size-oblivious version, the tester only gets ϵ as an explicit input.

Definition 2.1 (property testing in the bounded-degree graph model): *A* tester for the graph property Π (in the standard version of the bounded-degree graph model) *is a probabilistic oracle machine T that, on input a* proximity parameter $\epsilon > 0$ *and* size parameter $|V|$, *and access to two oracles, an incidence function $g : V \times [d] \to V \cup \{\perp\}$ and a device denoted* $\texttt{Samp}(V)$ *that samples uniformly in V, satisfies the following two conditions:*

1. *The tester accepts each graph $G = (V, E)$ in Π with probability at least $2/3$; that is, for every $g : V \times [d] \to V \cup \{\perp\}$ representing a graph in Π* (and $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\texttt{Samp}(V)}(|V|, \epsilon) = 1] \geq 2/3$.

2. *Given $\epsilon > 0$ and oracle access to any graph G that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $g : V \times [d] \to V \cup \{\perp\}$ that represents a graph that is ϵ-far from Π, it holds that $\Pr[T^{g,\texttt{Samp}(V)}(|V|, \epsilon) = 0] \geq 2/3$, where the graph represented by $g : V \times [d] \to V \cup \{\perp\}$ is* ϵ-far from Π *if for every $g' : V \times [d] \to V \cup \{\perp\}$ that represents a graph in Π it holds that $|\{(v, j) \in V \times [d] : g(v, j) \neq g'(v, j)\}| > \epsilon \cdot d|V|$.*

The tester is said to have one-sided error probability *if it always accepts graphs in Π; that is, for every $g : V \times [d] \to V \cup \{\perp\}$ representing a graph in Π* (and $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\texttt{Samp}(V)}(|V|, \epsilon) = 1] =$

[6] For simplicity, we adopt the standard convention by which the neighbors of v appear in arbitrary order in the sequence $(g(v, 1), \ldots, g(v, \deg(v)))$, where $\deg(v) \stackrel{\text{def}}{=} |\{j \in [d] : g(v, j) \neq \perp\}|$.

1. *The* size-oblivious version *is obtained by not providing the tester with $|V|$; that is, a* size-oblivious tester *is only given the proximity parameter $\epsilon > 0$* (and access to the oracles $g : V \times [n] \to V \cup \{\perp\}$ and $\texttt{Samp}(V)$).

The query complexity of a tester for Π is a function (of the parameters d, n and ϵ) that represents the number of queries made by the tester on the worst-case n-vertex graph of maximum degree d, when given the proximity parameter ϵ. Fixing d, we typically ignore its effect on the complexity (equiv., treat d as a hidden constant). Our focus in this section will be on cases in which the query complexity depends both on the proximity parameter and the size parameter, but we start with the case in which the complexity depends only on the proximity parameter (i.e., size-oblivious query complexity).

Open Problem 2.2 (classes of properties of size-oblivious query complexity): *Characterize the class of properties that can be tested within complexity that depends only on the proximity parameter.*

Several natural properties that admit testers of size-oblivious query complexity are surveyed in [13, Sec. 9.2 and 9.5]. In particular, subgraph freeness, degree regularity, connectivity, and cycle-freeness can be tested in query complexity $\text{poly}(1/\epsilon)$. In addition, a recent result of [31] asserts that all minor-closed properties can be tested in query complexity $\text{poly}(1/\epsilon)$. Indeed, it is hard to find a common theme among these properties, but still one may suggest the following problem.

Open Problem 2.3 (the class of properties of $\text{poly}(1/\epsilon)$ query complexity): *Provide a characterization of the class of properties that can be tested using* $\text{poly}(1/\epsilon)$ *queries.*

As in the dense graph model, of special interest (and possibly easier to analyze) are the special cases of query complexity $\widetilde{O}(1/\epsilon^2)$ and $\widetilde{O}(1/\epsilon)$.

Turning to size-dependent query complexity, we wonder whether sublinear query complexity is possible for testing graph isomorphism. We actually, refer to two version of the problem: The (massively parametrized [32]) version in which one is required to test for isomorphism to a fixed large graph, and the version of testing isomorphism between two input graphs.

Open Problem 2.4 (testing graph isomorphism): *What is the query complexity of testing isomorphism to a fixed large graph? Ditto for testing isomorphism between two input graphs.*

Recall that the following lower bounds are known for these two versions [17].

1. The query complexity of testing isomorphism to a fixed n-vertex graph is $\widetilde{\Omega}(n^{1/2})$.
2. The query complexity of testing isomorphism between two n-vertex graphs is $\widetilde{\Omega}(n^{2/3})$.

The lower bounds are shown by using graphs that have connected components of size $\text{poly}(\log n)$, and in this case the lower bounds are tight (up to a multiplicative factor of $\text{poly}(\log n)/\epsilon^2$) [17]. A seemingly related problem is that of testing the set of graphs that have only a trivial automorphism (a.k.a asymmetric graphs).

Open Problem 2.5 (testing asymmetric graphs): *What is the query complexity of testing the set of asymmetric graph?*

Some partial results are presented in [18].

One-sided vs two-sided error testing. As in the case of the dense graph model, we wonder if something systematic and appealing can be said about this subject. An interesting example of a property having a qualitative formulation and a quantitative approximation that seems responsible for the gap between the complexity of one-sided and two-sided error testing is cycle-freeness (see [21, 8]). See also [30, 31], which refer to any minor-free properties.

Other levels of query complexity. We mention that, although known hierarchy theorems establish the existence of properties with quite arbitrary query complexity (for testing) [20, 14], identifying natural properties of different query complexity levels is an interesting challenge. Of particular interest are complexities of the type $Q_1(n, \epsilon) = \text{poly}(\epsilon^{-1} \log n)$, $Q_2(n, \epsilon) = \exp(1/\epsilon)$, and $Q_3(n, \epsilon) = O(n^c \cdot F(1/\epsilon))$ for arbitrary $c \in (0, 0.5) \cup (0.5, 1)$ and $F : \mathbb{N} \to \mathbb{N}$. (Indeed, the values of $c \in \{0, 0.5, 1\}$ are excluded, since for these values we do know of natural properties.)

A recently resolved open problem. In relation to Problem 2.2, a recent work [1] disproves a favorite conjecture of ours that asserted that all "generalized subgraph freeness" properties (see [24, Def. 5.1]) have proximity-oblivious tester, which in turn implies that they could be tested within complexity that depends only on the proximity parameter. In fact, Adler, Kohler, and Peng [1] prove that some generalized subgraph freeness properties are not testable within complexity that depends only on the proximity parameter.

The notion of *generalized subgraph freeness* is aimed to capture what one can see by exploring a constant-radius neighborhood of a vertex in a graph. The issue is that some vertices are fully explored (i.e., one sees all their neighbors), whereas for others (at the boundary of the exploration) one may only encounter them but not all their neighbors. Formally, we consider marked graphs, which are graphs in which each vertex is marked either full or partial. *Such a marked graph $H = ([h], F)$ can be* embedded *in a graph $G = ([N], E)$ if there exists a 1-1 mapping $\phi : [h] \to [N]$ such that for every $i \in [h]$ the following two conditions hold:*

1. *If i is marked full, then ϕ yields a bijection between the set of neighbors of i in H and the set of neighbors of $\phi(i)$ in G. That is, in this case $\Gamma_G(\phi(i)) = \phi(\Gamma_H(i))$, where $\Gamma_X(v)$ denotes the set of neighbors of v in the graph X, and $\phi(S) = \{\phi(v) : v \in S\}$.*

2. *If i is marked partial, then ϕ yields an injection of the set of neighbors of i in H to the set of neighbors of $\phi(i)$ in G. That is, in this case $\Gamma_G(\phi(i)) \supseteq \phi(\Gamma_H(i))$.*

*The graph G is called H-*free *if H cannot be embedded in G* (i.e., there is no embedding of H in G that satisfies the foregoing conditions)*. For a set of marked graphs $\mathcal{F}$, a graph G is called $\mathcal{F}$-*free *if for every $F \in \mathcal{F}$ the graph G is F-free.*[7]

Indeed, the standard notion of (non-induced) subgraph freeness is a special case of generalized subgraph freeness, obtained by considering the corresponding marked graph in which all vertices are marked partial.[8] Introducing vertices that are marked full adds a new type of constraint;

[7]The original formulation of [24, Def. 5.1], allows also a third marking, called semi-full, with the semantics that if i is marked semi-full, then $\Gamma_G(\phi(i)) \cap \phi([h]) = \phi(\Gamma_H(i))$ (i.e., ϕ yields a bijection between the set of neighbors of i in H and the set of neighbors of $\phi(i)$ in the subgraph of G induced by $\phi([h])$). However, as pointed out in [24], this seemingly more expressive formulation is actually equivalent to the one presented here.

[8]Similarly, the notion of induced subgraph freeness is a special case of generalized subgraph freeness; this is easiest to see when allowing also semi-full marking (see Footnote 7), since then the effect of induced subgraph freeness is obtained by using marked graph in which all vertices are marked semi-full.

specifically, this constraint mandates the non-existence of neighbors that are outside the marked subgraph.[9] While [1] proves that generalized subgraph freeness properties may not be tested within size-oblivious complexity, it leaves open the problem of determining the query complexity of testing each of these properties.

3 Testing Graph Properties in the General Graph Model

Here we consider the standard and the size-oblivious versions of the notion of property testing in the general graph model (which was introduced in [34, 29] and is reviewed in [13, Chap. 10]).

Unlike in the previous two models, here the representation of the graph is decoupled from the definition of the (*relative*) distance between graphs. Following the discussion in [13, Sec. 10.1.2], we define the relative distance between $G = (V, E)$ and $G' = (V, E')$ as the ratio of the symmetric difference of E and E' over $\max(|E|, |E'|) + |V|$.

In this model, a graph $G = (V, E)$ is redundantly represented by both its incidence function $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ (alternatively, we may consider $g_1 : V \times [b(|V|)] \to V \cup \{\perp\}$, where $b : \mathbb{N} \to \mathbb{N}$ is some degree-bounding function)[10] and its adjacency predicate $g_2 : V \times V \to \{0, 1\}$; indeed, as before, $g_1(v, j) = u \in V$ if u is the j^{th} neighbor of v (and $g_1(v, j) = \perp$ if v has less than j neighbors), and $g_2(u, v) = 1$ if and only if $\{u, v\} \in E$. The tester is given oracle access to the two representations of the input graph (i.e., to the functions g_1 and g_2) as well as to a device that returns uniformly distributed elements in the graph's vertex-set. As usual, the tester is also given the proximity parameter ϵ. In addition, in the standard version, the tester is also given the size of the vertex-set as auxiliary input. Hence, the explicit inputs given to it are $|V|$ and ϵ. In the size-oblivious version, the tester only gets ϵ as an explicit input.

Definition 3.1 (property testing in the general graph model, revised):[11] *A* tester for the graph property Π (in the standard version of the general graph model) *is a probabilistic oracle machine T that, on input ϵ and $|V|$, and access to three oracles, an incidence function $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$, an adjacency predicate $g_2 : V \times V \to \{0, 1\}$, and a device denoted* `Samp`(V) *that samples uniformly in V, satisfies the following two conditions:*

1. *The tester accepts each graph $G = (V, E)$ in Π with probability at least $2/3$; that is, for every $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ and $g_2 : V \times V \to V \cup \{\perp\}$ representing a graph in Π* (and $\epsilon > 0$)*, it holds that* $\Pr[T^{g_1, g_2, \mathtt{Samp}(V)}(|V|, \epsilon) = 1] \geq 2/3$.

2. *Given $\epsilon > 0$ and oracle access to any graph G that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and (g_1, g_2) such that $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ and $g_2 : V \times V \to V \cup \{\perp\}$ represent a graph that is ϵ-far from Π, it holds that* $\Pr[T^{g_1, g_2, \mathtt{Samp}(V)}(|V|, \epsilon) = 0] \geq 2/3$, *where the graph $G = (V, E)$ is ϵ-*far from Π *if for every*

[9] For example, using vertices that are marked full, it is possible to disallow certain degrees in the graph. Thus, the generalized notion of subgraph freeness includes properties that are not hereditary (e.g., regular graphs), whereas induced and non-induced subgraph freeness are hereditary.

[10] In the first version of [15], we considered $g_1 : V \times [|V| - 1] \to V \cup \{\perp\}$, where $|V| - 1$ served as a trivial degree bound. In retrospect, we feel that using such an upper bound is problematic, because it may allow the tester to determine the number of vertices in the graph (assuming that querying g_1 on an input that is not in its domain results in a suitable indication). On the other hand, allowing an infinite representation of finite graphs is not problematic, because the representation is not used as a basis for the definition of the relative distance between graphs.

[11] Here we follow [13, Def. 10.2], rather than [13, Def. 10.1]. See discussion in [13, Sec. 10.1.2].

$G' = (V, E')$ that represents a graph in Π it holds that the symmetric difference of E and E' is greater than $\epsilon \cdot (\max(|E|, |E'|) + |V|)$.

The tester is said to have one-sided error probability *if it always accepts graphs in Π. The* size-oblivious version *is obtained by not providing the tester with $|V|$.*

The query complexity of a tester for Π is a function (of the parameters n and ϵ) that represents the number of queries made by the tester on the worst-case n-vertex graph, when given the proximity parameter ϵ. It is also common to consider the query complexity as a function of additional parameters such as the average degree of the graph's vertices or the graph's arboricity.

3.1 Preliminary thoughts

Two pivotal (related) problems in this model are estimating the average degree of the graph and selecting random edges in it. Denoting the average vertex degree by $\overline{d}$, both problems have query (and time) complexity $(\widetilde{\Theta}(n)/\overline{d})^{1/2}$; see [13, Sec. 10.3]. Note the more accurate picture that is offered by considering the effect of $\overline{d}$ (in contrast to the $\widetilde{\Theta}(n^{1/2})$ bounds (which hold when only assuming $\overline{d} = \Omega(1)$)). Furthermore, when taking also the arboricity into account, one gets upper bounds of the form $\alpha \cdot \text{poly}(\log n)/\overline{d}$, where $\alpha \in [\overline{d}, (\overline{d} \cdot n)^{1/2}]$ denotes the graph's arboricity [10]. The foregoing difference manifests itself when considering the complexity of testing cycle-freeness (see Appendix B).

Taking the arboricity into account is one way of limiting the vast variety of graphs that may occur in this general model. Even when fixing one's attention to sparse graphs (say, of average constant degree), there is a fundamental difference between graphs in which all vertices have approximately the same degree and graphs in which vertex degrees vary extremely (e.g., some vertices have degree $\Omega(n^{1/2})$, whereas almost all other vertices have constant degree). Needless to say, global properties may behave very differently in these two cases, frustrating attempts to present a single result that holds for all cases.

We mention, however, that in some cases the aforementioned differences do not matter. One such example is testing Bipartitness: Ignoring the polynomial dependence on $1/\epsilon$, this problem has query complexity $\widetilde{\Theta}(n^{1/2})$ both in the bounded-degree graph model [21, 22] and in the general graph model [29], when assuming $\overline{d} = O(n^{1/2})$. (For $\overline{d} = \Omega(n^{1/2})$, the complexity is $(\widetilde{\Theta}(n)/\overline{d})$ [29].)

3.2 So what are the questions?

It seems that the first question to pose is what should be considered reasonable goals. For sure, testing in the general graph model can only be harder than in the bounded-degree graph model, and the first question is when can it actually be as good. In light of the above, we cannot expect this to happen for general graphs, but it may be possible for graphs of bounded arboricity.

Open Problem 3.2 (from bounded degree to bounded arboricity): *Suppose that property Π is testable within complexity $Q(n, \epsilon) > 0$ in the bounded-degree graph model.*[12] *Provide an upper bound on the complexity of testing Π in the general graph model under the promise that the tested graph has constant arboricity. For example, can the latter complexity be linear in $Q(n, \epsilon)$, while permitting extra* $\text{poly}(\log n)$ *or $1/\epsilon$ factors?*

[12]The lower-bound on Q is aimed to prevent trivial counterexamples (i.e., properties that always or never hold in a bounded-degree graph).

Valuable partial answers may refer to all natural graph properties for which sub-linear testing results are known in the bounded-degree graph model. A counterexample may also be valuable, alas less pleasing.

Appendix A: On deriving size-oblivious testers

In the appendix, we show that *some known results* that seem to rely on providing the tester with the size of the graph can actually be emulated by size-oblivious testers (with two-sided error).

A.1 In the Dense Graph Model

Recall that the celebrated testers for subgraph-freeness [3] and generalized partition problems [19] are actually size-oblivious. In contrast, the almost-trivial testers presented in [13, Sec.8.2.2] seem to require knowledge of the size of the tested graph in order to determine their operation (i.e., whether or not to explore the entire graph). We show that this knowledge can be gained by sampling, and derive size-oblivious versions of the foregoing testers.

Specifically, as discussed in [13, Sec.8.2.2], graph properties can be almost-trivial to test either because every graph is close to the property (e.g., Hamiltonicity) or because the property contains only sparse graphs (e.g., Planarity). In both cases, trivial (or almost trivial) testing is possible when the proximity parameter is larger that the covering-radius or the sparsity parameter, which are stated in terms of the number of vertices in the graph, and otherwise one can afford to explore the entire graph. We convert these testers to size-oblivious ones by using a sufficiently large (in terms of ϵ) sample of vertices to decide in which situation we are, and act accordingly. Since this decision is probabilistic, we end-up having two-sided error probability.

Theorem A.1 (almost-trivially due to small covering-radius of the property, a size-oblivious version of [13, Prop. 8.3]): *Let Π be a graph property and $c \in (0,2)$. If every n-vertex graph is n^{-c}-close to Π, then there exists a size-oblivious tester of query complexity $(1/\epsilon)^{2/c}$ for Π.*

Proof: On input $\epsilon > 0$ and access to the graph $G = (V,E)$, we first check whether or not $\epsilon \geq |V|^{-c}$ (equiv., $|V| \geq (1/\epsilon)^{1/c}$). We do so by taking a sample of $\widetilde{O}(1/\epsilon)^{1/c})$ vertices and deciding positively if we see at least $t \stackrel{\rm def}{=} (1/\epsilon)^{1/c}$ vertices. In the latter case (i.e., seeing at least t vertices), we accept without making any other queries. Otherwise (i.e., seeing less than t vertices), we query all the vertex-pairs, and decide according to whether the induced subgraph is in Π.

If $\epsilon \geq |V|^{-c}$, then (w.h.p.) the sample contain at least $(1/\epsilon)^{1/c}$ vertices, which implies that $\epsilon \geq |V|^{-c}$, which in turn implies that G is ϵ-close to Π and so that we did well in accepting it. On the other hand, if $\epsilon < |V|^{-c}$, then (w.h.p.) the sample contains all vertices of V, and in this case we retrieve the graph G and decide correctly. ■

Theorem A.2 (almost-trivially due to sparseness of graphs in the property, a size-oblivious version of [13, Prop. 8.4]): *Let Π be a graph property and $c \in (0,2)$. If every n-vertex graph in Π has at most n^c edges, then there exists a size-oblivious tester of query complexity $(3/\epsilon)^{2/(2-c)}$ for Π.*

Proof Sketch: On input $\epsilon > 0$ and access to the graph $G = (V,E)$, we first check whether or not $\epsilon > 3|V|^{-(2-c)}$ (equiv., $|V| > (3/\epsilon)^{1/(2-c)}$). We do so by taking a sample of $\widetilde{O}(1/\epsilon)^{1/(2-c)})$ vertices and proceed according to whether or not we saw more than $t \stackrel{\rm def}{=} (3/\epsilon)^{1/(2-c)}$ vertices. If we saw more than t vertices, then we use $O(1/\epsilon)$ random queries to estimate the edge density of the graph such that it distinguishes between density at least $2\epsilon/3$ and density at most $\epsilon/3$. In the first case, we rejects (since the graph is not sufficiently sparse), and in the second case we accepts (since the graph is close enough to the empty graph, which is close enough to Π).[13] Otherwise (i.e., if we saw less than t vertices), we query all the vertex-pairs, and decide according to whether the induced subgraph is in Π. ■

A.2 In the Bounded-Degree Graph Model

While almost-trivial testers of natural graph properties are less common in the bounded-degree graph model (than in the dense graph model), such result do exist (and rely on the hypothesis that the tester is given the number of vertices in the tested graph).[14] We can adapt such testers to the size-oblivious version by using ideas as in Section A.1.

A more interesting issue is that of testers in the standard version that use the number of vertices in order to determine their operation. The archetypical example is the Bipartiteness tester of [22]. Recall that when testing an n-vertex graph, this tester picks $O(1/\epsilon)$ random vertices, takes $\text{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{n})$ random walks of length $\text{poly}(\epsilon^{-1} \log n)$ from each of these vertices, and accepts if and only if it sees no odd-length cycle (in the explored subgraph). It is quite evident that this tester works as well also when obtaining only a constant factor approximation of n, whereas such an approximation can be obtained by using $O(\sqrt{n})$ random vertices. The point is that we don't have to determine the number of samples beforehand; we rather keep sampling the vertex set untill we seen enough collisions. Hence, we have

Theorem A.3 (size-oblivious version of the Bipartite tester): *There exists a size-oblivious tester for Bipartiteness that runs in expected* $\text{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{n})$*-time. Furthermore, for any* $t \in \mathbb{N}$*, the tester runs at most* $t \cdot \text{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{n})$*-time.*

Proof Sketch: We keep taking random samples (of the vertex set) till the number of collisions is at least five. At this point, we set n to be the square of the number of samples used, and invoke the Bipartiteness tester of [22] with this value of n.

Recall that the collision probability of sampling uniformly from the vertex-set V is $1/|V|$, and so the expected number of pairwise collisions seen among m samples is $\binom{m}{2}/|V|$. Observing that the probability that n is set to be smaller than $|V|$ is upper-bounded by the probability of seeing at least five pairwise collisions among $\sqrt{|V|-1}$ samples, we upper-bound the former probability by

$$\frac{\binom{\sqrt{|V|-1}}{2}}{5 \cdot |V|} < 0.1.$$

On the other hand, using the fact that the sample-pairs are almost pairwise independent, one may show that, with probability at least 0.9, it holds that $O(\sqrt{|V|})$ samples contain at least five pairwise collisions, and in that case n is set to $O(|V|)$. Furthermore, for any $t > 1$, the probability that the tester runs t times longer than expected (equiv., n is set to $O(t^2 \cdot |V|)$) is at most $\exp(-t)$.[15] ■

[13]If the edge density is higher than $\epsilon/3$, then G has more than $(\epsilon/3) \cdot |V|^2 > |V|^c$ edges, and $G \notin \Pi$ follows. On the other hand, if the edge density is smaller than $2\epsilon/3$, then G is $2\epsilon/3$-close to the empty graph, which implies that G is ϵ-close to Π (since the empty graph is $\epsilon/3$-close to Π).

[14]One example is the almost-trivial tester for symmetric graphs [18, end of Sec. 2].

[15]To see this, partition the sample to t parts, and consider the probability that less than five pairwise collisions occur in each part.

Appendix B: On testing cycle freeness in the general graph model

Testing cycle-freeness (with two-sided error) in the bounded-degree graph model amounts to estimating the number of edges and the number of connected components (and comparing these two estimates), where in the latter case we may focus on small connected components (see [13, Sec. 9.2.5]). The simple procedure for estimating the number of small connected components is easily adaptable to the general graph model (cf. [13, Sec. 10.2.1]). Specifically, we may select $t = O(1/\epsilon^2)$ start vertices uniformly at random, and start an exploration of the graph at each of them, while suspending the exploration once $s = O(1/\epsilon)$ vertices are encountered. The number of connected components is estimated as $(1/t) \cdot \sum_{i\in[t]}(n/n_i)$, where $n_i \leq s+1$ is the number of vertices encountered in the $i^{\rm th}$ exploration.[16]

This leaves us with the task of estimating the number of edges, which (as briefly discussed in Section 3.1) is a pivotal task in the general graph model. As stated in Section 3.1, this task can be performed in time $\alpha \cdot \mathrm{poly}(\epsilon^{-1}\log n)/\overline{d}$, where $\overline{d}$ denotes the average vertex degree, and $\alpha \in [\overline{d}, (\overline{d}\cdot n)^{1/2}]$ denotes the arboricity of the graph. We stress that in the general case $\alpha/\overline{d} = O(n/\overline{d})^{1/2}$, whereas $\alpha/\overline{d} = O(1)$ when $\alpha = O(\overline{d})$. This gap is manifested in the complexity of the testing problem.

Proposition B.1 (testing cycle freeness in the general graph model): *Assuming that* $\overline{d}/\epsilon = \mathrm{poly}(\log n)$*, the complexity of testing cycle freeness is* $\widetilde{\Theta}(n^{1/2})$ *in the general case and* $\mathrm{poly}(\log n)$ *in the bounded arboricity case.*

The lower bound in the general case can be demonstrated by observing that $\Omega(n^{1/2})$ queries are needed to distinguish between an n-vertex graph that consists of a tree on $n - n^{1/2}$ vertices and $n^{1/2}$ isolated vertices and the same tree augmented by an $n^{1/2}$-vertex clique.

References

[1] I. Adler, N. Kohler, and P. Peng. GSF-locality is not sufficient for proximity-oblivious testing. GSF-locality is not sufficient for proximity-oblivious testing. In *36th CCC*, 2021.

[2] N. Alon. Testing subgraphs of large graphs. *Random Structures and Algorithms*, Vol. 21, pages 359–370, 2002.

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

[4] 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.

[5] N. Alon and A. Shapira. A Separation Theorem in Property Testing. *Combinatorica*, Vol. 28 (3), pages 261–281, 2008.

[16]Letting s_v denote the size of the connected component in which vertex v reside, note that the number of connected components in the graph equals $\mathrm{Exp}_v[n/s_v]$, since each connected component contributes one unit to the expected value.

[6] L. Avigad and O. Goldreich. Testing Graph Blow-Up. In *Studies in Complexity and Cryptography*, pages 156–172, 2011.

[7] M. Balcan, E. Blais, A. Blum, and L. Yang. Active Property Testing. In *53rd FOCS*, pages 21–30, 2012.

[8] A. Czumaj, O. Goldreich, D. Ron, C. Seshadhri, A. Shapira, and C. Sohler. Finding cycles and trees in sublinear time. *Random Structures & Algorithms*, Vol. 45(2), pages 139–184, 2014.

[9] A. Czumaj and C. Sohler. A Characterization of Graph Properties Testable for General Planar Graphs with one-Sided Error (It's all About Forbidden Subgraphs). In *60th FOCS*, pages 1525–1548, 2019

[10] T. Eden, D. Ron, and C. Seshadhri. Sublinear Time Estimation of Degree Distribution Moments: The Arboricity Connection. *SIAM Journal on Disc. Math. and Alg.*, Vol. 33 (4), pages 2267–2285, 2019.

[11] E. Fischer and A. Matsliah. Testing Graph Isomorphism. In *17th SODA*, pages 299–308, 2006.

[12] J. Fox. A New Proof of the Graph Removal Lemma. *Annals of Mathematics*, Vol. 174 (1), pages 561–579, 2011.

[13] O. Goldreich. *Introduction to Property Testing.* Cambridge University Press, 2017.

[14] O. Goldreich. Hierarchy Theorems for Testing Properties in Size-Oblivious Query Complexity. *Computational Complexity*, Vol. 28 (4), pages 709–747, 2019.

[15] O. Goldreich. Flexible models for testing graph properties. *ECCC*, TR18-104, 2018.

[16] O. Goldreich. Testing Graphs in Vertex-Distribution-Free Models. In *51st STOC*, pages 527–534, 2019

[17] O. Goldreich. Testing Isomorphism in the Bounded-Degree Graph Model. *ECCC*, TR19-102, 2019.

[18] O. Goldreich. On Testing Asymmetry in the Bounded Degree Graph Model. *ECCC*, TR20-118, 2020.

[19] 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.

[20] O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg. Hierarchy Theorems for Property Testing. *Computational Complexity*, Vol. 21 (1), pages 129–192, 2012.

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

[22] O. Goldreich and D. Ron. A Sublinear Bipartitness Tester for Bounded Degree Graphs. *Combinatorica*, Vol. 19 (3), pages 335–373, 1999.

[23] O. Goldreich and D. Ron. Algorithmic Aspects of Property Testing in the Dense Graphs Model. *SIAM Journal on Computing*, Vol. 40, No. 2, pages 376–445, 2011.

[24] O. Goldreich and D. Ron. On Proximity Oblivious Testing. *SIAM Journal on Computing*, Vol. 40, No. 2, pages 534–566, 2011.

[25] O. Goldreich and D. Ron. On Sample-Based Testers. In *6th ITCS*, pages 337–345, 2015.

[26] O. Goldreich and L. Trevisan. Three Theorems Regarding Testing Graph Properties. *Random Structures and Algorithms*, Vol. 23 (1), pages 23–57, August 2003.

[27] O. Goldreich and L. Trevisan. Errata to [26]. Manuscript, August 2005. Available from `http://www.wisdom.weizmann.ac.il/~oded/p_ttt.html`

[28] O. Goldreich and A. Wigderson. Non-adaptive vs Adaptive Queries in the Dense Graph Testing Model. *ECCC*, TR20-160, 2020.

[29] T. Kaufman, M. Krivelevich, and D. Ron. Tight Bounds for Testing Bipartiteness in General Graphs. *SIAM Journal on Computing*, Vol. 33 (6), pages 1441–1483, 2004.

[30] A. Kumar, C. Seshadhri, and A. Stolman. Finding Forbidden Minors in Sublinear Time: A $n^{1/2+o(1)}$-Query One-Sided Tester for Minor Closed Properties on Bounded Degree Graphs. In *59th FOCS*, pages 509–520, 2018.

[31] A. Kumar, C. Seshadhri, and A. Stolman. Random Walks and Forbidden Minors II: A poly$(d\epsilon^{-1})$-query Tester for Minor-Closed Properties of Bounded Degree Graphs. In *51st STOC*, pages 559–567, 2019.

[32] I. Newman. Property Testing of Massively Parametrized Problems – A Survey. In *Property Testing: Current Research and Surveys*, pages 142–157, Springer, LNCS, Vol. 6390, 2010.

[33] K. Onak and X. Sun. The Query Complexity of Graph Isomorphism: Bypassing Distribution Testing Lower Bounds. In *50th STOC*, pages 165–171, 2018.

[34] M. Parnas and D. Ron. Testing the Diameter of Graphs. *Random Structures and Algorithms*, Vol. 20 (2), pages 165–183, 2002.

[35] M. Parnas, D. Ron, and R. Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Science*, Vol. 72(6), pages 1012–1042, 2006.

[36] R. Rubinfeld and M. Sudan. Robust Characterization of Polynomials with Applications to Program Testing. *SIAM Journal on Computing*, 25(2), pages 252–271, 1996.