Electronic Colloquium on Computational Complexity, Report No. 184 (2022)

Testing in the bounded-degree graph model with degree bound two*

Oded Goldreich and Laliv Tauber
Department of Computer Science
Weizmann Institute of Science, Rehovot, Israel.

December 28, 2022

Abstract

Considering the bounded-degree graph model, we show that if the degree bound is two, then every graph property can be tested within query complexity that only depends on the proximity parameter. Specifically, the query complexity is $\text{poly}(1/\epsilon)$, where ϵ denotes the proximity parameter. The key observation is that a graph of maximum degree two consists of a collection of paths and cycles, and that a collection of long paths and cycles is relatively close (in this model) to a single cycle.

1 Introduction

This note refers to testing graph properties in the bounded-degree graph model, where graphs are represented by their incidence lists. This model was introduced in [5], and is surveyed in [1, Chap. 9]. Even when considering only natural graph properties, the query complexity of testing in this model varies from being independent of the size of the graph to being linear in it (see [1, Sec. 9.6]). When allowing also unnatural properties, the complexity can be practically anything (see [4, 2]). In contrast to the richness of these results, which hold even for degree bound 3, we show that the situation with degree bound 2 is quite dull. Specifically, we show that, when the degree bound is two, every graph property can be tested within query complexity $\text{poly}(1/\epsilon)$, where ϵ denotes the proximity parameter.

1.1 Background

The bounded-degree graph model refers to a fixed degree bound, denoted $d \geq 2$. An n-vertex graph $G = ([n], E)$, of maximum degree d, is represented in this model by a function $g : [n] \times [d] \to \{0, 1, ..., n\}$ such that $g(v, i) = u \in [n] \stackrel{\text{def}}{=} \{1, 2, ..., n\}$ if u is the i^{th} neighbor of v and $g(v, i) = 0$ if v has less than i neighbors. Hence, it is also adequate to refer to this model as the incidence function model. For simplicity, we assume here that the neighbors of vertex v appear in an arbitrary order in the sequence $g(v, 1), ..., g(v, \deg(v))$, where $\deg(v) \stackrel{\text{def}}{=} |\{i : g(v, i) \neq 0\}|$ is the degree of v. Also, we shall always assume that if $g(v, i) = u \in [n]$, then there exists $j \in [d]$ such that $g(u, j) = v$.

*Partially supported by the Israel Science Foundation (grant No. 1041/18) and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702).

ISSN 1433-8092

Distance between graphs is measured in terms of their aforementioned representation; that is, as the fraction of (the number of) different array entries over $d \cdot n$ (equiv., fraction of different edges over $dn/2$). We are interested in *graph properties*, which are sets of graphs that are closed under isomorphism; that is, Π is a graph property if for every graph $G=([n],E)$ and every permutation π of $[n]$ it holds that $G \in \Pi$ if and only if $\pi(G) \in \Pi$, where $\pi(G) \stackrel{\rm def}{=} ([n], \{\{\pi(u),\pi(v)\} : \{u,v\} \in E\})$. With these preliminaries in place, we now recall the meaning of property testing in this model.

Definition 1 (testing graph properties in the bounded-degree graph model): *For a fixed d, a* tester *for a graph property Π is a probabilistic oracle machine that, on input parameters n and ϵ, and access to* (the incidence function of) *an n-vertex graph $G = ([n], E)$ of maximum degree d, 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 $G' = ([n], E') \in \Pi$ of maximum degree d it holds that the symmetric difference between E and E' has cardinality that is greater than $\epsilon \cdot dn/2$.* (Equivalently, we may say that G is ϵ-far from G' if for every $g : [n] \times [d] \to \{0,1,...,n\}$ and $g' : [n] \times [d] \to \{0,1,...,n\}$ that represent G and G', respectively, it holds that $|\{(v,i) : g(v,i) \neq g'(v,i)\}| > \epsilon \cdot dn$.)[1]

If the tester accepts every graph in Π with probability 1, then we say that it has one-sided error*; otherwise, we say that it has* two-sided error*.*

The query complexity of a tester is the number of queries it makes to any n-vertex graph, as a function of the parameters n and ϵ.

1.2 Our results

Our main result (i.e., Theorem 5) is that, *when the degree bound equals 2, any graph property can be tested using* $\mathrm{poly}(1/\epsilon)$ *queries* (in the bounded-degree graph model). The proof of this result is based on two simple observations:

1. A graph of maximum degree two consists of a collection of paths and cycles.
2. Each m-vertex graph that consists of paths and cycles of length at least ℓ is $1/\ell$-close to a single m-vertex path (resp., cycle).

Combining these two observations, when given oracle access to an n-vertex graph (of maximum degree 2), we can "approximately learn" the graph (up to isomorphism) by approximating the number of i-vertex cycles and paths, for each $i \in [O(1/\epsilon)]$. Given this approximation of the graph, one can readily determine its distance to any graph property. Likewise, one can determine whether two graphs are isomorphic (or far from being so).

Organization. In Section 2 we present a relatively straightforward implementation of the foregoing ideas, which yields testers of query complexity $O(1/\epsilon^4)$. A more sophisticated implementation, which yields an almost optimal bound of $\widetilde{O}(1/\epsilon^2)$, is presented in Section 3.

[1]We say that G is ϵ-close to G' if G is not ϵ-far from G'; that is, $|\{(v,i) : g(v,i) \neq g'(v,i)\}| \leq \epsilon \cdot dn$.

2 Main results

Starting from the foregoing two observations, we associate graphs with the numbers of vertices in each type of connected component, where the main types correspond to paths and cycles of specific small lengths (e.g., 4-vertex paths and 7-vertex cycles). In addition, we also consider isolated vertices and connected components of large size. Approximating the number of vertices that reside in connected components of each of these types is done by sampling vertices and exploring the connected component in which they reside while stopping the exploration when detecting that the connected component is larger than some designated bound (denoted ℓ).

Specifically, for $\ell = O(1/\epsilon)$, we consider 2ℓ types of connected components. The types correspond to possible lengths of paths and cycles, where we distinguish each of the lengths up to ℓ (i.e., $(i+1)$-vertex paths and $(i+2)$-vertex cycles for $i \in [\ell-1]$) from isolated vertices (viewed as length-zero paths) and from paths and cycles of length greater than ℓ. Specifically,

- Type 1 correspond to isolated vertices;
- For $i \in [\ell-1]$ Type $2i$ corresponds to $(i+1)$-vertex paths, whereas Type $2i+1$ corresponds $(i+2)$-vertex cycles; and
- Type 2ℓ corresponds to either paths or cycles of length greater than in the other types (i.e., paths having more than ℓ vertices or cycles having more than $\ell+1$ vertices).

Note that a connected component with more than $\ell+1$ vertices can be detected by making at most $2\cdot(\ell+1)$ queries, and ditto for detecting an $(\ell+1)$-vertex path. For a graph $G = ([n], E)$, we let g_i denote the fraction of vertices that reside in connected components of Type i, where $i \in [2\ell]$, and call the sequence of g_i's the ℓ-profile of G.

Algorithm 2 (main algorithm): *On input $n \in \mathbb{N}$ and $\epsilon > 0$, given oracle access to a graph $G = ([n], E)$, letting $\ell = O(1/\epsilon)$, we sample $O(\ell/\epsilon^2)$ vertices, and determine for each sampled vertex the type of connected component in which it resides, where the type of each sample vertex is determined by making at most $2\cdot(\ell+1)$ queries* (per each sampled vertex). *We output $(\widetilde{g}_1, ..., \widetilde{g}_{2\ell})$, such that $\widetilde{g}_i$ denotes the fraction of sampled vertices that reside in connected components of Type i.*

The query (and time) complexity of this algorithm is $O(\ell^2/\epsilon^2) = O(1/\epsilon^4)$. Recall that a distribution over $[m]$ can be learned up to a total variation distance of ϵ by using $O(m/\epsilon^2)$ samples (see, e.g., [1, Exer. 11.4]). Letting the g_i's denote the actual ℓ-profile of the graph G, it follows that, *with* (say) *probability at least* 0.9*, it holds that*

$$\sum_{i \in [2\ell]} |\widetilde{g}_i - g_i| < \epsilon.$$

Note that the $\widetilde{g}_i$'s do not necessarily fit the profile of any n-vertex graph (e.g., they may not even be integer multiples of $1/n$). However, we shall consider and only make statements about the set of graphs that have an ℓ-profile that is close to $(\widetilde{g}_1, ..., \widetilde{g}_{2\ell})$. Specifically, we prove that if the ℓ-profiles of two graphs are at distance at most ϵ, then they are $O(\epsilon)$-close to being isomorphic.

Lemma 3 (profiles versus isomorphism): *For $\epsilon > 0$ and $\ell \geq 4/\epsilon$, let $(g_1, ..., g_{2\ell})$ be the ℓ-profile of $G = ([n], E)$ and $(g'_1, ..., g'_{2\ell})$ be the ℓ-profile of $G' = ([n], E')$. If $\sum_{i \in [2\ell]} |g_i - g'_i| \leq \epsilon$, then G is 2ϵ-close to being isomorphic to G'.*

Proof: We shall show that both G and G' are each ϵ-close to a graph with ℓ-profile $(g''_1, ..., g''_{2\ell})$ such that $g''_i = \min(g_i, g'_i)$ for every $i \in [2\ell - 1]$ (and $g''_{2\ell} = 1 - \sum_{i\in[2\ell-1]} g''_i$). Furthermore, this graph will contain a single connected component of Type 2ℓ, which will be a cycle. Although these two graphs (with ℓ-profile $(g''_1, ..., g''_{2\ell})$) may be different, they are isomorphic to one another. Hence, we focus on proving that G is ϵ-close to a graph G'' that satisfies the foregoing conditions (and the agrument for G' is analogous).

We first observe that

$$\sum_{i\in[2\ell-1]} |g_i - \min(g_i, g'_i)| \;\leq\; \sum_{i\in[2\ell]:g_i>g'_i} (g_i - g'_i) \;=\; \frac{1}{2} \cdot \sum_{i\in[2\ell]} |g_i - g'_i|,$$

where equality holds if $g_{2\ell} \leq g'_{2\ell}$. Hence, $\sum_{i\in[2\ell-1]} |g_i - g''_i| \leq 0.5 \cdot \epsilon$.

Let V_i denote the set of vertices of G that reside in connected component of Type i, and note that $|V_i| = g_i \cdot n$. For each $i \in [2\ell - 1]$, we let V''_i be an adequate $g''_i \cdot n$-vertex subset of V_i, and let $V''_{2\ell} = V_{2\ell} \cup \bigcup_{i\in[2\ell-1]} (V_i \setminus V''_i)$. Specifically, for each $i \in [2\ell - 1]$, the vertices in V''_i are complete connected components of G; that is, the movement does not split connected components of G. (This is the case because both $g_i \cdot n$ and $g''_i \cdot n$ are multiples of the number of vertices in each connected componet of Type i.)

Indeed, we aim at making V''_i be the set of vertices that reside in connected component of Type i in G''. Towards this end, we only need to modify the adjacencies of the vertices in $V''_{2\ell}$. The key observation is that the subgraph (of G) induced by $V''_{2\ell}$ can be made a single cycle by modifying the adjacencies of at most two vertices in each connected component in the subgraph (of G) induced by $V_{2\ell} \cup \bigcup_{i\in[2\ell-1]} (V_i \setminus V''_i)$. Recalling that connected components in $V_{2\ell}$ have size greater than ℓ, it follows that we only need to modify the adjacencies of $\frac{2}{\ell} \cdot |V_{2\ell}| + \sum_{i\in[2\ell-1]} (g_i - g''_i) \cdot n$ vertices. Using $|V_{2\ell}| \leq n$, $\ell \geq 4/\epsilon$ and $\sum_{i\in[2\ell-1]} |g_i - g''_i| \leq 0.5 \cdot \epsilon$, the claim follows. ■

Applications. Using Algorithm 2 and Lemma 3, we derive the testing results stated in the introduction. We start with the problem of testing isomorphism between a pair of input graphs, which requires a straightforward extension of the testing model.

Theorem 4 (testing isomorphism): *In the bounded-degree graph model with degree bound 2, testing isomorphism between two input graphs can be done in time $O(1/\epsilon^4)$.*

Proof: The tester consists of approximating the profiles of both input graphs by invoking Algorithm 2, with proximity parameter $\epsilon/8$, and accepting if and only if the difference between the two estimated profiles is at most $\epsilon/4$.

If the graphs are isomorphic, then, with probability at least 0.8, the estimated profiles are at distance at most $2 \cdot \epsilon/8$ from one another. On the other hand, if the graphs are ϵ-far from being isomorphic, then (by Lemma 3) their profiles are at distance greater than $\epsilon/2$ from one another. In that case, with probability at least 0.8, their estimated profiles are at distance greater than $\frac{\epsilon}{2} - 2 \cdot \frac{\epsilon}{8} = \epsilon/4$ from one another. ■

Theorem 5 (testing any graph property): *Let Π be an arbitrary graph property. Then, in the bounded-degree graph model with degree bound 2, the property Π can be tested using $O(1/\epsilon^4)$ queries.*

Proof: The tester consists of approximating the profile of the input graph by invoking Algorithm 2, with proximity parameter $\epsilon/4$, and accepting if and only if the difference between the estimated profile and the profile of some graph in the property is at most $\epsilon/4$.

Evidently, if the input graph is in Π, then its profile fits a profile of a graph in Π, and the tester will accept with probability at least 0.9. On the other hand, if the graph is ϵ-far from any graph in the property, then (by Lemma 3) its profile is at distance greater than $\epsilon/2$ from the profile of any graph in the property. In that case, with probability at least 0.9, its estimated profile is at distance greater than $\frac{\epsilon}{2} - \frac{\epsilon}{4} = \epsilon/4$ from any such profile. ■

3 Limitations and Improvements

The proofs of both Theorems 4 and 5 yield two-sided error testers. As noted in [3, Thm. 2.5], two-sided error is inherent to testing isomorphism with sub-linear query complexity in this model (even with degree bound 2). This holds even when testing isomorphism to a fixed n-vertex graph consisting of $n/6$ isolated triangles and $n/6$ length 2 paths.[2] Hence, two-sided error in also inherent to Theorem 5.

The aforementioned graph also demonstrates that both testing tasks require $\Omega(1/\epsilon^2)$ queries.[3] Furthermore, it seems that $\Omega(1/\epsilon^4)$ queries are needed in order to obtain an approximated profile in the sense obtained by Algorithm 2 (i.e., $(\widetilde{g}_1, ..., \widetilde{g}_{2\ell})$ such that $\sum_{i\in[2\ell]} |\widetilde{g}_i - g_i| \le \epsilon$, where $(g_1, ..., g_{2\ell})$ is the ℓ-profile of the input graph and $\ell = \Theta(1/\epsilon)$).[4] However, such an approximated profile is not required in order to perform the foregoing testing tasks. As argued next, it suffices to provide less accurate approximations of the number of vertices that belong to larger connected components.

The key observation is that the cost of moving vertices from connected components of one type to components of a different type is inversely proportional to the size of the smaller connected components (e.g., modifying m paths that are each of length ℓ' into a cycle of length $m\ell'$ can be achieved by $2m$ adjacency modification). On the other hand, the cost of determining the type of a connected component is upper bounded by its size, and so we can afford to provide a better approximation for the number of vertices in them.

The simpler implementation of the foregoing idea uses a *single threshold* (of $\sqrt{\ell}$), and distinguishes between small connected components (which have size smaller than this threshold) and larger ones. A more sophisticated implementation clusters the connected components according to their approximated size, and uses $O(\log \ell)$ such *clusters* (rather than two). These two implementations are presented in the following two subsections.

3.1 The single size-threshold algorithm

We focus on the design and analysis of a more efficient algorithm that replaces Algorithm 2. As before, we set $\ell = O(1/\epsilon)$. We use $t = \sqrt{\ell}$ as the threshold distinguishing the first $2t-1$ types from the remaining $2\ell - 2t + 1$ types. We approximate the first g_i's by $\widetilde{g}_i$'s such that $\sum_{i\in[2t-1]} |\widetilde{g}_i - g_i| < \epsilon$,

[2] Actually, we can establish the claim also with respect to degree bound 1, by using a fixed n-vertex graph consisting of $n/2$ isolated vertices and $n/4$ isolated edges.

[3] Consider the task of testing whether an input graph is isomorphic to the aforementioned fixed graph, denoted F. Then, $\Omega(1/\epsilon^2)$ queries are required to distinguish a random isomorphic copy of F from a random n-vertex graph that consists of $(1 + 3\epsilon)n/6$ isolated triangles and $(1 - 3\epsilon)n/6$ length 2 paths.

[4] Specifically, learning an unknown distribution over $[m]$ up to a total variation distance of ϵ requires $\Omega(m/\epsilon^2)$ samples, whereas determining the type of a connected component (wrt types in $[2\ell]$) requires $\Omega(\ell)$ queries.

using a sample of size $O(t/\epsilon^2)$, at the cost of making $O(t)$ queries per sampled vertex. In contrast, the remaining g_i's is approximated by $\widetilde{g}_i$'s such that $\sum_{i\in[2t,2\ell]}|\widetilde{g}_i - g_i| < t\cdot\epsilon$, using a sample of size $O(\ell/(t\cdot\epsilon)^2)$, at the cost of making $O(\ell)$ queries per sampled vertex. Specifically, the algorithm proceeds as follows.

1. We sample $O(t/\epsilon^2)$ vertices and determine $\widetilde{g}_1, ..., \widetilde{g}_{2t-1}$ by making at most $2\cdot(t+1)$ queries per each sampled vertex. As before (i.e., in Algorithm 2), $\widetilde{g}_i$ denotes the fraction of sampled vertices that reside in connected components of Type i.

2. Letting $\eta = t\cdot\epsilon$, we sample $O(\ell/\eta^2)$ vertices and determine $\widetilde{g}_{2t}, ..., \widetilde{g}_{2\ell}$ by making at most $2\cdot(\ell+1)$ queries (per each sampled vertex).

The query (and time) complexity of this algorithm is $O(t^2/\epsilon^2)+O(\ell^2/\eta^2) = O(1/\epsilon^3)$, since $t^2 = \ell = O(1/\epsilon)$. Letting the g_i's denote the actual profile of the input graph G, and using [1, Exer. 11.4], it follows that, *with* (say) *probability at least* 0.9, *it holds that* $\sum_{i\in[2t-1]}|\widetilde{g}_i - g_i| < \epsilon$ *and* $\sum_{i\in[2t,2\ell]}|\widetilde{g}_i - g_i| < \eta$. Again, we shall consider the set of graphs that have an ℓ-profile that is close to $(\widetilde{g}_1, ..., \widetilde{g}_{2\ell})$, but the notion of "being close" is adapted to fit the approximation provided by the foregoing algorithm.

Lemma 6 (a variant on Lemma 3): *For $\epsilon > 0$, $\ell = 2/\epsilon$ and $t = \sqrt{\ell}$, let $(g_1, ..., g_{2\ell})$ be the ℓ-profile of $G = ([n], E)$ and $(g'_1, ..., g'_{2\ell})$ be the ℓ-profile of $G' = ([n], E')$. If $\sum_{i\in[2t-1]}|g_i - g'_i| \le \epsilon$, $\sum_{i\in[2t,2\ell]}|g_i - g'_i| \le \eta \stackrel{\rm def}{=} t\cdot\epsilon$, then G is 8ϵ-close to being isomorphic to G'.*

Using Lemma 6, we obtain improvements over the complexity bounds stated in Theorems 4 and 5. Specifically, the bound is improved from $O(1/\epsilon^4)$ to $O(1/\epsilon^3)$.

Proof: Intuitively, when moving vertices between two different types that correspond to connected components of size greater than t it suffices to modify the adjacencies of a $2/t$ fraction of the vertices. Hence, the second sum (i.e., $\sum_{i\in[2t,2\ell]}|g_i - g'_i|$), which is smaller than η, contributes only $O(\eta/t) = O(\epsilon)$ units to the difference between G and G'. The contribution of the first sum (i.e., $\sum_{i\in[2t-1]}|g_i - g'_i|$), remains $O(\epsilon)$ as before. Details follow.

As in the proof of Lemma 3, we shall actually prove that each of the two graphs (i.e., G and G') is 4ϵ-close to being isomorphic to a graph G'' that has the ℓ-profile $(g''_1, ..., g''_{2\ell})$ such that $g''_i = \min(g_i, g'_i)$ for every $i \in [2\ell-1]$ (and $g''_{2\ell} = 1 - \sum_{i\in[2\ell-1]} g''_i$). Furthermore, G'' will contain a single connected component of Type 2ℓ, which will be a cycle. The construction of G'' is identical to the one in the proof of Lemma 3, and all that changes is the analysis of the number of vertices whose adjacency is modified.

Recall that we upper-bounded this number by two vertices per each connected component in the subgraph (of G) induced by $V_{2\ell} \cup \bigcup_{i\in[2\ell-1]}(V_i \setminus V''_i)$. Here, we use the fact that each of the connected components in the subgraph induced by $\bigcup_{i\in[2t,2\ell-1]}(V_i \setminus V''_i)$ have size greater than t. It follows that we only need to modify the adjacencies of at most

$$\frac{2}{\ell}\cdot|V_{2\ell}| + \sum_{i\in[2t-1]}(g_i - g''_i)\cdot n + \frac{2}{t}\cdot\sum_{i\in[2t,2\ell-1]}(g_i - g''_i)\cdot n$$

vertices. Using $\sum_{i\in[2t-1]}|g_i - g''_i| \le \epsilon$ and $\sum_{i\in[2t,2\ell-1]}|g_i - g''_i| \le t\cdot\epsilon$ (as well as $|V_{2\ell}| \le n$ and $\ell \ge 2/\epsilon$), the claim follows. ■

3.2 The size-clustering algorithm

As before, we set $\ell = O(1/\epsilon)$, but here ℓ is a power of 2. We partition $[2\ell-1]$ into $\ell' = \log_2(2\ell)$ intervals, denoted $I_1, ..., I_{\ell'}$, such that $I_k = [2^{k-1}, 2^k - 1]$. Our main goal will be to approximate the g_i's by $\widetilde{g}_i$'s such that $\sum_{i \in I_k} |\widetilde{g}_i - g_i| < \epsilon_k \stackrel{\rm def}{=} 2^{k-1} \cdot \epsilon$ for every $k \in [\ell']$. (Note that $|\widetilde{g}_{2\ell} - g_{2\ell}| < 2^{\ell'-1} \cdot \epsilon$ always holds, since we shall use $\ell > 1/\epsilon$.) For each $k \in [\ell']$, we approximate the $(g_i)_{i \in I_k}$'s by sampling $O(t \cdot |I_k|/\epsilon_k^2)$ vertices, making at most $2 \cdot ((2^k - 1) + 1) = 2^{k+1}$ queries per each sampled vertex, and determining the $\widetilde{g}_i$'s for each $i \in I_k$; specifically, we use the following algorithm.

Algorithm 7 (main algorithm, revisited): *On input $n \in \mathbb{N}$ and $\epsilon > 0$, given oracle access to a graph $G = ([n], E)$, we set $\ell = 2^{\ell'-1} = O(1/\epsilon)$ and $t = \log_2(10\ell')$. For each $k \in [\ell']$, recalling that $I_k = [2^{k-1}, 2^k - 1]$ and $\epsilon_k = 2^{k-1} \cdot \epsilon$, we proceed as follows.*

- *We sample $O(t \cdot |I_k|/\epsilon_k^2)$ vertices. For each sampled vertex, making at most 2^{k+1} queries, we determine whether it resides in a connected component with type in I_k, and if so the type of this connected component.*
- *For each $i \in I_k$, we let $\widetilde{g}_i$ denote the fraction of sampled vertices that reside in connected components of Type i.*

We output $(\widetilde{g}_1, ..., \widetilde{g}_{2\ell})$, where $\widetilde{g}_{2\ell} \leftarrow 1 - \sum_{i \in [2\ell-1]} \widetilde{g}_i$.

Hence, for each k, with probability at least $1 - 2^{-t}$, it holds that $\sum_{i \in I_k} |\widetilde{g}_i - g_i| < \epsilon_k$, and using $t = \log_2(10\ell')$ implies that all ℓ' inequalities hold with probability at least 0.9. The query (and time) complexity of Algorithm 7 is

$$\sum_{k \in [\ell']} O(t \cdot |I_k|/\epsilon_k^2) \cdot O(2^k) \;=\; \sum_{k \in [\ell']} O(t \cdot 2^{2k}/(2^{k-1} \cdot \epsilon)^2) \;=\; O(t \cdot \ell'/\epsilon^2) \;=\; \widetilde{O}(\ell')/\epsilon^2.$$

Again, we shall consider the set of graphs that have an ℓ-profile that is close to $(\widetilde{g}_1, ..., \widetilde{g}_{2\ell})$, but the notion of "being close" is adapted to fit the approximation provided by Algorithm 7.

Lemma 8 (another variant on Lemma 3): *For $\epsilon > 0$, $\ell' = \lceil \log_2(4/\epsilon) \rceil$ and $\ell = 2^{\ell'-1}$, let $I_k = [2^{k-1}, 2^k - 1]$ and $\epsilon_k = 2^{k-1} \cdot \epsilon$. If $\sum_{i \in I_k} |g_i - g'_i| \le \epsilon_k$ for every $k \in [\ell']$, where $\overline{g} = (g_1, ..., g_{2\ell})$ is the ℓ-profile of $G = ([n], E)$ and $\overline{g}' = (g'_1, ..., g'_{2\ell})$ is the ℓ-profile of $G' = ([n], E')$, then G is $O(\ell' \cdot \epsilon)$-close to being isomorphic to G'.*

Proof: We generalized the proof of Lemma 6, using again the same construction of an intermediate graph G'' as in the proof of Lemma 3, and adapting the analysis to the current setting. That is, we shall show that each of the two graphs (i.e., G and G') is $O(\ell' \cdot \epsilon)$-close to being isomorphic to a graph G'' that has the ℓ-profile $(g''_1, ..., g''_{2\ell})$ such that $g''_i = \min(g_i, g'_i)$ for every $i \in [2\ell-1]$ (and $g''_{2\ell} = 1 - \sum_{i \in [2\ell-1]} g''_i$). (Again, G'' will contain a single connected component of Type 2ℓ, which will be a cycle.)

Recall that we upper-bounded the number of vertices whose adjacencies is modified by two vertices per each connected component in the subgraph (of G) induced by $V_{2\ell} \cup \bigcup_{i \in [2\ell-1]} (V_i \setminus V''_i)$. Here we use the fact that the connected components in the subgraph induced by $\bigcup_{i \in I_k} (V_i \setminus V''_i)$ have size greater than 2^{k-2}. It follows that we only need to modify the adjacencies of at most

$$\frac{2}{\ell} \cdot |V_{2\ell}| + \sum_{k \in [\ell']} \frac{2}{2^{k-2}} \cdot \sum_{i \in I_k} (g_i - g''_i) \cdot n$$

vertices. Using $\sum_{i\in I_k} |g_i - g_i''| \leq \epsilon_k = 2^{k-1} \cdot \epsilon$ (as well as $|V_{2\ell}| \leq n$ and $\ell \geq 2/\epsilon$), we derive an upper bound of $\epsilon \cdot n + \sum_{k\in[\ell']} 4\epsilon \cdot n$, and the claim follows. ■

Theorem 9 (testing isomorphism, revisited): *In the bounded-degree graph model with degree bound 2, testing isomorphism between two input graphs can be done in time $\widetilde{O}(1/\epsilon^2)$.*

Proof: The tester consists of approximating the profiles of both input graphs by invoking Algorithm 7, with proximity parameter $\epsilon' \stackrel{\rm def}{=} \epsilon/O(\log(1/\epsilon))$, and accepting if and only if for every k it holds that the (norm-1) difference between the approximated frequencies that correspond to types in I_k is at most $2^k \cdot \epsilon'$. Note that the time (and query) complexity of this tester is $\widetilde{O}(\log(1/\epsilon')) \cdot (1/\epsilon')^2 = \widetilde{O}(\log^3(1/\epsilon)) \cdot (1/\epsilon)^2$.

If the graphs are isomorphic, then, with probability at least 0.8, the estimated frequencies for each k are at distance at most $2 \cdot 2^{k-1} \cdot \epsilon'$, and the tester accepts. On the other hand, if the graphs are ϵ-far from being isomorphic, then (by Lemma 8), for some k, the difference between the actual frequencies that corresponding to types in I_k is greater than $4 \cdot 2^{k-1} \cdot \epsilon'$. In that case, the corresponding estimates are at distance greater than $4 \cdot 2^{k-1} \cdot \epsilon' - 2 \cdot 2^{k-1} \cdot \epsilon'$ (with probability at least 0.8). ■

Theorem 10 (testing any graph property, revisited): *Let Π be an arbitrary graph property. Then, in the bounded-degree graph model with degree bound 2, the property Π can be tested using $\widetilde{O}(1/\epsilon^2)$ queries.*

Proof: The tester consists of approximating the profile of the input graph G by invoking Algorithm 7, with proximity parameter $\epsilon' = \epsilon/O(\log(1/\epsilon))$, and accepting if and only if there exists a graph G' in Π such that for every k it holds that the (norm-1) difference between the frequencies (of G' and G) that correspond to types in I_k is at most $2^{k-1} \cdot \epsilon'$.

Evidently, if the input graph is in Π, then its profile fits a profile of a graph in Π, and the tester will accept with probability at least 0.9. On the other hand, if the graph G is ϵ-far from any graph G' in the property, then (by Lemma 8), for some k, the difference between the actual frequencies (of G and G') that corresponding to types in I_k is greater than $2 \cdot 2^{k-1} \cdot \epsilon'$. In that case, the corresponding estimates of the frequencies of G are at distance greater than $2 \cdot 2^{k-1} \cdot \epsilon' - 2^{k-1} \cdot \epsilon'$ (with probability at least 0.9, regardless of G'). ■

References

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

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

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

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

[5] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, pages 302–343, 2002. Extended abstract in *29th STOC*, 1997.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il