The complexity of testing all properties of planar graphs, and the role of isomorphism

Sabyasachi Basu[*]   Akash Kumar[†]   C. Seshadhri[‡]

August 23, 2021

Abstract

Consider property testing on bounded degree graphs and let $\varepsilon > 0$ denote the proximity parameter. A remarkable theorem of Newman-Sohler (SICOMP 2013) asserts that *all* properties of planar graphs (more generally hyperfinite) are testable with query complexity only depending on ε. Recent advances in testing minor-freeness have proven that all additive and monotone properties of planar graphs can be tested in $\mathrm{poly}(\varepsilon^{-1})$ queries. Some properties falling outside this class, such as Hamiltonicity, also have a similar complexity for planar graphs. Motivated by these results, we ask: can all properties of planar graphs can be tested in $\mathrm{poly}(\varepsilon^{-1})$ queries? Is there a uniform query complexity upper bound for all planar properties, and what is the "hardest" such property to test?

We discover a surprisingly clean and optimal answer. Any property of bounded degree planar graphs can be tested in $\exp(O(\varepsilon^{-2}))$ queries. Moreover, there is a matching lower bound, up to constant factors in the exponent. The natural property of testing isomorphism to a fixed graph requires $\exp(\Omega(\varepsilon^{-2}))$ queries, thereby showing that (up to polynomial dependencies) isomorphism to an explicit fixed graph is the hardest property of planar graphs. The upper bound is a straightforward adapation of the Newman-Sohler analysis that tracks dependencies on ε more carefully. The main technical contribution is the lower bound construction, which is achieved by a special family of planar graphs that are all mutually far from each other.

We can also apply our techniques to get analogous results for bounded treewidth graphs. We prove that all properties of bounded treewidth graphs can be tested in $\exp(O(\varepsilon^{-1} \log \varepsilon^{-1}))$ queries. Moreover, testing isomorphism to a fixed forest requires $\exp(\Omega(\varepsilon^{-1}))$ queries.

1 Introduction

Consider the setting of property testing for bounded degree graphs, under the model of random access to a graph adjacency list, as introduced by Goldreich-Ron [GR02]. Let $G = (V, E)$ be a graph where $V = [n]$ and the maximum degree is d. We have random access to the list through

---

[*]Department of Computer Science, University of California, Santa Cruz. sbasu3@ucsc.edu

[†]School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne. akash.kumar@epfl.ch This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 759471)

[‡]Department of Computer Science, University of California, Santa Cruz. sesh@ucsc.edu

neighbor queries. There is an oracle that, given $v \in V$ and $i \in [d]$, returns the ith neighbor of v (if no neighbor exists, it returns $\perp$).

For a property $\mathcal{P}$ of graphs with degree bound d, the distance of G to $\mathcal{P}$ is the minimum number of edge additions/removals required to make G have $\mathcal{P}$, divided by dn. We say that G is ε-far from $\mathcal{P}$ if the distance to $\mathcal{P}$ is more than ε. A property tester for $\mathcal{P}$ is a randomized procedure that takes as input (query access to) G and a proximity parameter, $\varepsilon > 0$. If $G \in \mathcal{P}$, the tester must accept with probability at least 2/3. If G is ε-far from $\mathcal{P}$, the tester must reject with probability at least 2/3. In our context, the property $\mathcal{P}$ is called *testable* if there exists a property tester for $\mathcal{P}$ whose query complexity is independent of n.

One of the grand goals of property testing is to classify testable (graph) properties according to the query complexity of testing them. Of special interest are *efficiently testable properties*, whose testing complexity is poly(ε^{-1}). For bounded degree graphs, there is little clarity on this issue. A recent survey by Goldreich states: "Indeed, it is hard to find a common theme among [efficiently testable] properties...". [Gol21]

Our work focuses on *properties of planar graphs*. Even the problem of just testing planarity has received much attention, whose complexity has only recently been shown to be poly(ε^{-1}) [BSS08, HKNO09, EHNO11, YI15, LR15, KSS19]. Newman-Sohler proved that every planar property (actually, every hyperfinite property) is testable, but they do not provide an explicit complexity bound depending on ε [NS13]. Kumar-Seshadhri-Stolman recently showed that every *additive and monotone* planar property can tested in poly(ε^{-1}) queries [KSS19]. Applying techniques from this result, Levi-Shoshan showed that planar Hamiltonicity can be tested efficiently [LS21]. The latter property is neither additive nor monotone, so it is natural to ask if all planar properties can be tested efficiently. If not, does there exist a uniform complexity bound for all planar properties, and a candidate for the "hardest planar property"? (We note that there is significant work on characterizing testable planar properties, for the *unbounded degree* case, which is qualitatively different [CMOS11, CS19]. Details are given in §1.2.)

We discover that the answer to this question is the query complexity bound $\exp(\Theta(\varepsilon^{-2}))$. Up to constant factors in the exponent, the hardest planar property is testing isomorphism to a fixed explicit graph.

Let us give some formalism, and discuss the connection to isomorphism. In our discussions, n is basically fixed, so we are considering a non-uniform setting. A *planar property* Π is a set of unlabeled bounded degree planar graph with n vertices. The input G is a labeled graph and the tester is trying to property test if G is isomorphic (or equal to, if one ignores the labels) to any member of Π. Observe that singleton properties, where $\Pi = \{H\}$, are equivalent to testing if G is isomorphic to an fixed graph H. With this premable, we can state our main results.

Our upper bound is a straightforward adaptation of arguments in Newman-Sohler [NS13].

Theorem 1.1. *Consider a planar property* Π *of n-vertex, d degree-bounded graphs. There is a property tester for* Π *that makes* poly$(d) \exp(O(\varepsilon^{-2}))$ *queries.*

Our main technical result is a matching lower bound for an explicit singleton property. One of the surprises (at least to the authors) is that testing isomorphism to an arbitrary set of planar graphs is not harder, up to polynomial dependencies, than testing isomorphism to a single fixed graph. A recent compendium of open problems by Goldreich states the question of determining the complexity of testing isomorphism (Open Problem 2.4 in [Gol21]). We note that our theorems resolve this question for bounded-degree planar graphs.

Theorem 1.2. *For every sufficiently large n, there exists a bounded-degree planar graph H on n vertices such that property testing $\Pi = \{H\}$ requires $\exp(\Omega(\varepsilon^{-2}))$ queries. Equivalently, testing isomorphism to H requires $\exp(\Omega(\varepsilon^{-2}))$ queries.*

Bounded treewidth classes: In the context of property testing, many results for planar graphs also hold for minor-free classes. It is natural to ask whether the bound of $\exp(\Theta(\varepsilon^{-2}))$ is the "right" answer for any property of minor-free graphs. We discover that to *not* be the case. For properties of bounded treewidth graphs, the answer is between $\exp((\varepsilon^{-1}))$ and $\exp((\varepsilon^{-1})\log\varepsilon^{-1})$. The following theorems are also derived from the same methods for planar graphs, but the constructions are substantially simpler. The lower bound is achieved by a simple construction of forests. The main point of these results is to show that the bounded of $\exp(\Theta(\varepsilon^{-2}))$ achieved for planar properties can be significantly beaten for non-trivial minor-closed families.

Theorem 1.3. *Consider a bounded treewidth property Π of n-vertex, d degree-bounded graphs. There is a property tester for Π that makes $\mathrm{poly}(d)\exp(O(\varepsilon^{-1}\log\varepsilon^{-1}))$ queries.*

Theorem 1.4. *For every sufficiently large n, there exists a bounded-degree forest H on n vertices such that property testing $\Pi = \{H\}$ requires $\exp(\Omega(\varepsilon^{-1}))$ queries.*

1.1 Main ideas

One of the key components of property testers for planar graphs is the notion of *partition oracles for hyperfinite graphs*, as introduced by Hassidim-Kelner-Nguyen-Onak [HKNO09]. Bounded degree planar graphs (and more generally, minor-closed families) are hyperfinite, meaning that one can remove a constant fraction of edges and obtain connected components of constant size. Specifically, one can remove εdn edges to get connected components of size $O(\varepsilon^{-2})$.

Suppose G is hyperfinite. A partition oracle is a local procedure that gives access to a hyperfinite decomposition to a graph G, without any preprocessing. Given a query vertex v, the partition oracle outputs a connected subgraph $C(v)$ of size $O(\varepsilon^{-2})$, such that the union of components $C(v)$ forms a hyperfinite partition. Recent results give a partition oracle that runs in time $\mathrm{poly}(\varepsilon^{-1})$ per query [LR15, KSS21].

Newman-Sohler used partition oracles to prove that all hyperfinite (and hence planar) properties are testable [NS13]. This is where our results begin. Stripping down the Newman-Sohler arguments to their core, we can essentially treat a planar G, up to εdn edge changes, as a distribution over connected planar graphs of size $O(\varepsilon^{-2})$. The partition oracle allows us to sample efficiently from this distribution. (If the input graph G is not hyperfinite, the partition oracle can detect that efficiently, and the graph can be directly rejected.) For convenience, let $\mathcal{D}(G)$ denote this distribution.

Existing theorems in combinatorics show that the number of planar graphs of size $O(\varepsilon^{-2})$ is $\exp(O(\varepsilon^{-2}))$ [MR01]; hence, this bounds the support size of $\mathcal{D}(G)$. We can learn an approximation of $\mathcal{D}(G)$ up to ε TV-distance with $\exp(O(\varepsilon^{-2}))$ queries to the partition oracle. It is not hard, but central to the argument, to show that if $\|\mathcal{D}(G) - \mathcal{D}(H)\|_1 \leq \varepsilon$, then G is ε-close to H. For any property Π, one can simply check exhaustively if the learned approximation to $\mathcal{D}(G)$ is close to $\mathcal{D}(H)$ for any $H \in \Pi$. (While this could be expensive in running time, it requires no further queries to G.)

The main challenge is in the lower bound. We need to find a property Π such that testing Π requires $\exp(\Omega(\varepsilon^{-2}))$ samples from $\mathcal{D}(G)$. We seem to need a converse to the upper bound

argument, showing if $\|\mathcal{D}(G) - \mathcal{D}(H)\|_1$ is large, then G and H are far from each other. But this is false in general! The support of $\mathcal{D}(G)$ is a set of graphs, which are mutable objects. Meaning, we can modify $\mathcal{D}(G)$ dramatically by only modifying a few edges of G. (As an extreme case, $\mathcal{D}(G)$ and $\mathcal{D}(H)$ could have disjoint supports on graphs that are close to each other.)

All our graphs (both input and hard instance) will be collections of connected components of size $O(\varepsilon^{-2})$. We begin by trying to construct a graph H such that: for all G, $\|\mathcal{D}(G) - \mathcal{D}(H)\|_1 \geq \varepsilon$ implies that G is ε-far from H. Moreover, determining if $\|\mathcal{D}(G) - \mathcal{D}(H)\|_1 \geq \varepsilon$ should require $\exp(\Omega(\varepsilon^{-2}))$ samples from $\mathcal{D}(G)$. A candidate is suggested by the problem of uniformity testing of distributions. Suppose we construct H where $\mathcal{D}(H)$ is a uniform distribution on a collection $\mathcal{F}$ of graphs (each member of which has size $O(\varepsilon^{-2})$), such that $|\mathcal{F}| = \exp(\Omega(\varepsilon^{-2}))$. By standard distribution testing arguments, distinguishing the uniform distribution on $\mathcal{F}$ from a uniform distribution on *half* of $\mathcal{F}$ requires $\sqrt{|\mathcal{F}|} = \exp(\Omega(\varepsilon^{-2}))$ samples. Suppose furthermore that all graphs in $\mathcal{F}$ are ε-far from each other and all balanced separators of $F \in \mathcal{F}$ are at least of size $\varepsilon|F|$. We prove that any graph G such that $\mathcal{D}(G)$ is supported on half of $\mathcal{F}$ must be ε-far from H. Moreover, we can construct candidate input graphs G, where any property tester can be simulated by sampling from $\mathcal{D}(G)$. Putting all the arguments together, we get a bonafide lower bound: property testing $\Pi = \{H\}$ requires $\exp(\Omega(\varepsilon^{-2}))$ queries.

What remains is the main technical construction. We need to design $\mathcal{F}$, the "suitable family" of graphs for the lower bound. We achieve this family by taking the $\varepsilon^{-1} \times \varepsilon^{-1}$ grid and adding a collection of diagonal edges. We apply a cleaning procedure to get a large family of graphs that are all far from each other. The separator size holds trivially, since each graph in $\mathcal{F}$ contains a large grid.

1.2 Related Work

Property testing on bounded-degree graphs is a vast topic and we direct the reader to Chapter 9 of Goldreich's book [Gol17] for an introduction to the subject.

Arguably, the starting point for testing planarity is the seminal work of Benjamini-Schramm-Shapira [BSS08], who showed that all minor-closed properties are testable. This paper also introduced the significance of hyperfiniteness to property testing. Hassidim-Kelner-Nguyen-Onak [HKNO09] introduced the concept of partition oracles, a key tool in property testing for hyperfinite classes. Improving the query time of partition oracles was addressed by Edelman-Hassidim-Nguyen-Onak [EHNO11], Levi-Ron [LR15] and Kumar-Seshadhri-Stolman [KSS21].

The quest for characterizing testable (bounded-degree) graph properties and finding properties testable in $\text{poly}(\varepsilon^{-1})$ is an important theme in property testing. We note that Goldreich's recent survey explicitly calls these out as Open Problems 2.2 and 2.3 [Gol21]. One of the main inspirations for our work is the result of Newman-Sohler that proves that *all* hyperfinite properties are testable [NS13]. The recent work of [KSS21] proves that all additive and monotone minor-closed properties are efficiently testable, and Levi-Shoshan show that Hamiltonicity of minor-closed families is also efficiently testable [LS21].

We note an important line of work of testing properties of planar graphs, in the *unbounded degree case*. This direction was pioneered by Czumaj-Monemizadeh-Onak-Sohler [CMOS11], who showed the bipartitenesss is testable (independent of the size). A recent result of Czumaj-Sohler prove that all testable planar properties in the unbounded degree setting are related to subgraph freeness [CS19].

2 Proof of Theorem 1.1 and Theorem 1.3

We begin with some preliminaries: the notions of hyperfiniteness and partition oracles. Note that all graphs have n vertices and degree bound d.

Definition 2.1. *A graph G is (ε, k) hyperfinite if there exists a subset of εdn edges whose removal results in connected components of size at most k.*

A classic result of Alon-Seymour-Thomas (Prop. 4.1 of [AST94]) shows that all minor-closed families are $(\varepsilon, O(\varepsilon^{-2}))$-hyperfinite. In Lemma A.3, we prove that all bounded treewidth families are $(\varepsilon, O(\varepsilon^{-1}))$-hyperfinite. A partition oracle gives local access to a hyperfinite decomposition. We give the formal definition below (adapted from Def. 1.1 of [KSS21]).

Definition 2.2. *A procedure $\boldsymbol{A}$ is a partition oracle for a minor-closed family Π of graphs if it satisfies the following properties. The deterministic procedure takes as input random access to $G = (V, E)$, access to a random seed r, a proximity parameter $\varepsilon > 0$, and a vertex v of G. (We will think of fixing G, r, ε, so we use the notation $\boldsymbol{A}_{G,r,\varepsilon}$. All probabilities are with respect to r.) The procedure $\boldsymbol{A}_{G,r,\varepsilon}(v)$ outputs a connected set of vertices, such that the sets $\{\boldsymbol{A}_{G,r,\varepsilon}(v)\}$, over all v, form a partition of V.*

We say that the partition oracle outputs an (ε, k)-hyperfinite decomposition if the following properties hold. (i) For all v, $|\boldsymbol{A}_{G,r,\varepsilon}(v)| \leq k$ and (ii) with probability $> 2/3$ (over r), the number of edges between the sets $\boldsymbol{A}_{G,r,\varepsilon}(v)$ is at most εdn.

The main result of [KSS21] that we use is the following.

Theorem 2.3 (Rephrasing of Theorem 1.2 [KSS21])**.** *For any bounded degree graph in a minor-closed family, there is a partition oracle that outputs an $(\varepsilon, O(\varepsilon^{-2}))$-hyperfinite partition, and runs in time $O(\mathrm{poly}(d\varepsilon^{-1}))$ per query.*

For bounded treewidth classes, the partition oracle can output $(\varepsilon, O(\varepsilon^{-1}))$-hyperfinite partitions using standard arguments (Claim A.4)

2.1 Property testers through subgraph count vectors

We use partition oracles to summarize a hyperfinite graph by an approximate count vector. This technique was first used by Newman-Sohler [NS13]. We follow their analysis, but take care to keep track of various dependencies on ε. This allows for getting the optimal query complexity bound.

Consider any planar property Π (technically, the following arguments hold for any minor-closed property). For any planar G, let $\mathcal{P}(G)$ denote the partition given by the partition oracle of Theorem 2.3. (Note that the partition is a random variable.) Let $\mathcal{B}_k$ be the set of unlabeled graphs in the family Π with at most k vertices. We will always set $k = O(\varepsilon^{-2})$ (though the exact setting will change for bounded-treewidth properties).

For any graph G' consisting of connected components of size at most k and any $F \in \mathcal{B}_k$, let $ct_F(G')$ be the number of occurrences of F in G'. Let $ct(G')$ be the $|\mathcal{B}_k|$-dimensional vector of these counts. Note that $\| ct(G') \|_1 \in [n/k, n]$.

Claim 2.4. *Consider two graphs G'_1, G'_2 consisting entirely of connected components of size at most k. If $\| ct(G'_1) - ct(G'_2) \|_1 \leq \gamma n$, then $dist(G'_1, G'_2) \leq \gamma k d$.*

Proof: For every $F \in \mathcal{B}_k$, we will modify G'_1 and G'_2 to equalize the $ct_F(G'_1)$ and $ct_F(G'_2)$. We simply delete $|ct_F(G'_1) - ct_F(G'_2)|$ instances from either G'_1 or G'_2 (whichever has the larger count). This operation deletes at most $|ct_F(G'_1) - ct_F(G'_2)|kd$ edges. In total, the number of edges deleted is at most $\|ct(G'_1) - ct(G'_2)\|_1 kd \leq \gamma nkd$. $\qquad\square$

For the property Π that we wish to test, construct the following set $\boldsymbol{C}_\Pi$ of count vectors: for every $F \in \Pi$ (with n vertices and degree bound d) and every subgraph F' that is an (ε, k) partition of F, add $ct(F')$ to $\boldsymbol{C}_\Pi$.

Lemma 2.5. *Suppose $\mathcal{P}(G)$ is a valid (ε, k)-partition of G. If $G \in \Pi$, then $ct(\mathcal{P}(G)) \in \boldsymbol{C}_\Pi$. If G is 3ε-far from Π, then $\forall \boldsymbol{v} \in \boldsymbol{C}_\Pi$, $\|ct(\mathcal{P}(G)) - \boldsymbol{v}\|_1 > \varepsilon n/kd$.*

Proof: Suppose $G \in \Pi$. In the construction of $\boldsymbol{C}_\Pi$ described above, we can select F as G and F' as $\mathcal{P}(G)$. Hence, $ct(\mathcal{P}(G)) \in \boldsymbol{C}_\Pi$.

Suppose G is 3ε-far from Π. Consider any $\boldsymbol{v} \in \boldsymbol{C}_\Pi$, so $\boldsymbol{v} = ct(F')$, for F' being an (ε, k)-partition of $F \in \Pi$. By triangle inequality, $dist(\mathcal{P}(G), F') \geq dist(G, F) - dist(G, \mathcal{P}(G)) - dist(F, F')$. By farness, $dist(G, F) \geq 3\varepsilon$. Because $\mathcal{P}(G)$ and F' are respective (ε, k)-partitions, $dist(G, \mathcal{P}(G))$ and $dist(F, F')$ are at most ε. Hence, $dist(\mathcal{P}(G), F') \geq \varepsilon$. By Claim 2.4, $\|ct(\mathcal{P}(G)) - \boldsymbol{v}\|_1 > \varepsilon n/kd$. $\qquad\square$

Now, we present the main algorithmic ingredient of our property tester.

Claim 2.6. *Given G and a setting of $\mathcal{P}(G)$ that is a valid (ε, k)-partition, using $O(\text{poly}(d\varepsilon^{-1})\delta^{-2}|\mathcal{B}_k|^3 \log|\mathcal{B}_k|)$ queries one can compute a vector $\boldsymbol{v}$ such that $\|\boldsymbol{v} - ct(\mathcal{P}(G))\|_1 < \delta n$ with probability at least $1 - \frac{1}{|\mathcal{B}_k|}$.*

Proof: Fix $F \in \mathcal{B}_k$. We show how to approximate $ct_F(\mathcal{P}(G))$. Pick uar vertex s, and using the partition oracle, determine the component of $\mathcal{P}(G)$ containing s. If the component is isomorphic to F, declare success. The probability of success is exactly $|F| ct_F(\mathcal{P}(G))/n$. By Chernoff-Hoeffding, we can get an additive $\delta/|\mathcal{B}_k|$ estimate with error probability $< |\mathcal{B}_k|^{-2}$ using $O(|\mathcal{B}_k|^2\delta^{-2}\log|\mathcal{B}_k|)$ samples. Thus, we get an additive $\delta n/(|F| \cdot |\mathcal{B}_k|) \leq \delta n/|\mathcal{B}_k|$ estimate for $ct_F(\mathcal{P}(G))$. Applying for all F, we get our estimate vector $\boldsymbol{v}$. Note that the total approximation is $\|\boldsymbol{v} - ct(\mathcal{P}(G))\|_1 < \delta n/|\mathcal{B}_k| \times |\mathcal{B}_k| = \delta n$. The error probability, by union bound, is at most $|\mathcal{B}_k|^{-1}$. By the running time bound of the partition oracle, the total number of queries made is $O(\text{poly}(d\varepsilon^{-1})\delta^{-2}|\mathcal{B}_k|^3\log|\mathcal{B}_k|)$. $\qquad\square$

Now, we prove Theorem 1.1, repeated for convenience. To get the optimal bound for planar properties, we need the right upper bound for $\mathcal{B}_k$. Numerous results in the past show that the answer is $\exp(O(k))$. We cite one specific reference.

Theorem 2.7. *(Theorem 5 of [MR01]) A planar graph on k vertices and k' edges can be represented uniquely by $8k + 2k' + o(k + k')$ bits.*

Hence, the number of planar graphs on k vertices is $\exp(O(k))$.

Theorem 1.1. *Consider a planar property Π of n-vertex, d degree-bounded graphs. There is a property tester for Π that makes $\text{poly}(d)\exp(O(\varepsilon^{-2}))$ queries.*

Proof: The tester has two phases. In the first phase, it uses Theorem 2.3 to get a hyperfinite partition. If it succeeds, then the tester proceeds to the second phase. It uses Claim 2.6 to approximate $ct(\mathcal{P}(G))$ and checks if it lies in $\boldsymbol{C}_\Pi$.

The tester repeats the following $O(1)$ times. For a random seed R, it sets up the partition oracle with this seed, and then estimates the number of cut edges via random sampling. This is

6

done by sampling $\Theta(\varepsilon^{-1})$ uar vertices u, picking a uar neighbor v of u, and calling the partition oracle on u and v. If the outputs are different components, then edge (u, v) is cut. If more than an $\varepsilon/4$-fraction of edges are cut, the process is repeated with a new choice of R. Otherwise, R is fixed, and the tester proceeds to the next phase. If no R is found, the tester rejects.

At this point, a suitable $\mathcal{P}(G)$ has been discovered. We set $k = O(\varepsilon^{-2})$, to be the size bound of components, as promised by Theorem 2.3. Using Claim 2.6, the tester computes an approximate count vector $\boldsymbol{v}$ of $ct(\mathcal{P}(G))$, setting $\delta = \varepsilon/4kd$. Note that $\boldsymbol{C}_\Pi$ is a fixed set of vectors, independent of the input. The tester then determines if $\exists \boldsymbol{w} \in \boldsymbol{C}_\Pi$ such that $\|\boldsymbol{v} - \boldsymbol{w}\|_1 \leq \varepsilon n/2kd$. If such a vector $\boldsymbol{w}$ exists, it accepts. Otherwise, it rejects. The description of the property tester is complete, and we proceed to the analysis.

Query complexity bound: Note that $k = O(\varepsilon^{-2})$, $\delta = \varepsilon/4kd$, and, by Theorem 2.7, $|\mathcal{B}_k| = \exp(O(\varepsilon^{-2}))$. Hence, by Claim 2.6, the overall query complexity is $\text{poly}(d)\exp(O(\varepsilon^{-2}))$.

Correctness analysis: If G is planar and hence $(\varepsilon/8, O(\varepsilon^{-2}))$-hyperfinite, with probability at least $2/3$ over R, the partition oracle computes a hyperfinite decomposition. Hence, with probability at least $5/6$, the tester proceeds to the second phase with $\mathcal{P}(G)$ being a valid $(\varepsilon, O(\varepsilon^{-2}))$-hyperfinite partition. Conversely, with probability at least $5/6$, if the tester finds a suitable R, then $\mathcal{P}(G)$ is a valid $(\varepsilon, O(\varepsilon^{-2}))$-hyperfinite partition.

Suppose $G \in \Pi$ and hence planar. As argued above, with probability $> 5/6$, $\mathcal{P}(G)$ is an $(\varepsilon, O(\varepsilon^{-2}))$-hyperfinite decomposition. By Claim 2.6, with probability $> 5/6$, $\|\boldsymbol{v} - ct(\mathcal{P}(G))\|_1 < \varepsilon n/4kd$. By the union bound, both conditions hold with probability at least $2/3$. By Lemma 2.5, $ct(\mathcal{P}(G)) \in \boldsymbol{C}_\Pi$. Thus, there exists $\boldsymbol{w}$ in $\boldsymbol{C}_\Pi$ such that $\|\boldsymbol{v} - \boldsymbol{w}\|_1 \leq \varepsilon n/2kd$ and the tester accepts with probability at least $2/3$.

Suppose G is 3ε-far from Π. By Lemma 2.5, for all $\boldsymbol{w} \in \boldsymbol{C}_\Pi$, $\|ct(\mathcal{P}(G)) - \boldsymbol{w}\|_1 > \varepsilon n/kd$. By the triangle inequality, $\|\boldsymbol{v} - \boldsymbol{w}\|_1 > \varepsilon n/kd - \varepsilon n/4kd > \varepsilon n/2kd$. Thus, the tester rejects with probability at least $2/3$.

$\square$

The proof of Theorem 1.3 is nearly identical.

Theorem 1.3. *Consider a bounded treewidth property Π of n-vertex, d degree-bounded graphs. There is a property tester for Π that makes $\text{poly}(d)\exp(O(\varepsilon^{-1}\log\varepsilon^{-1}))$ queries.*

Proof: We redo the proof of Theorem 1.1. Claim A.4 shows that graphs with treewidth at most τ admit a partition oracle with $k = O(\tau/\varepsilon)$. For bounded treewidth graphs, we apply the trivial bound $|\mathcal{B}_k| \leq 2^{O(k\log k)}$ (Claim A.1). We apply these bounds in the above proof, and get a query complexity of $O(\text{poly}(d)\exp(\varepsilon^{-1}\log\varepsilon^{-1}))$ queries. $\square$

3 Lower bounds through suitable families

The key construct for the lower bounds is given in the following definition. A graph class is monotone if it is closed under edge removals. A graph class is additive if it closed under taking disjoint unions of graphs.

Definition 3.1. *Fix $\varepsilon > 0$ and a monotone, additive graph class $\mathcal{C}$. We call a family $\mathcal{F}$ of distinct graphs (in $\mathcal{C}$) ε-suitable if the following conditions hold:*
- *All graphs in $\mathcal{F}$ have the same number of vertices (denoted t).*
- *If $t > 2/\varepsilon$, $\forall F, F' \in \mathcal{F}$, $dist(F, F') \geq 0.02$.*

- $\forall F \in \mathcal{F}$, the size of any $(\frac{0.01}{d}, 1 - \frac{0.01}{d})$ balanced separator in F is $\Omega(\varepsilon t)$.

The main lemma used in our lower bound says that a large suitable family leads to a property testing lower bound.

Lemma 3.2. *Fix $\varepsilon > 0$, and let $\mathcal{F}$ denote an ε-suitable family of a monotone, additive graph class $\mathcal{C}$. Then, for all sufficiently large n, there exists a graph $H \in \mathcal{C}$ with the following property. Any property tester for the property $\{H\}$ with proximity parameter ε requires $\Omega(\sqrt{|\mathcal{F}|})$ queries.*

The proof of Lemma 3.2 requires some tools, that we shall build up in this section. Firstly, using the properties of a suitable family, we can construct two graphs consisting entirely of components in $\mathcal{F}$ that are far from each other. These graphs form the core of the lower bound.

Claim 3.3. *Let $\mathcal{F}$ denote an ε-suitable family of a graph class $\mathcal{C}$ (as defined in Lemma 3.2). Let $I = [|\mathcal{F}|]$ denote a set indexing graphs in $\mathcal{F}$.*

Let H_1 be the graph consisting of a disjoint union of all graphs in $|\mathcal{F}|$. For any subset $R \subseteq I$ with $|R| = |I|/2$, let H_R be the disjoint union of two copies of each graph in $\mathcal{F}$ indexed by R. Then the graphs H and H_R are $\Omega(\varepsilon)$ far from each other.

Proof: Since $\mathcal{F}$ is ε-suitable, we know that all graphs $F \in \mathcal{F}$ are defined on $t = t(\varepsilon)$ vertices. We know for any $F, F' \in \mathcal{F}$, since $|F| = |F'| = t$ the number of edge edits needed to change F to F' is at least $\Omega(t)$. Since R will be fixed for the remainder of the proof, it will be convenient to refer to H_R as H_2 for the rest of the argument. Let us denote the components of H_1 (resp H_2) as $\texttt{components(H}_1\texttt{)}$ (resp as $\texttt{components(H}_2\texttt{)}$). Let $s = |\texttt{components}(\mathcal{F})|$. H_2 contains $s/2$ connected components in $\mathcal{F}$ each of which occurs twice. Recall from Definition 3.1, $C, C' \in \texttt{components}(\mathcal{F})$ are $\Omega(1)$-far from each other. Let $\mathcal{M}_{1/2}$ denote the set of *missing* components of $\mathcal{F}$ in H_2. And let $\mathcal{P}_{1/2}$ denote the set of components from $\mathcal{F}$ that are *present* in H_2 (without duplicates). Thus, H_2 contains two subgraphs isomorphic to $\mathcal{P}_{1/2}$ which we denote as $\mathcal{P}^1_{1/2}$ and $\mathcal{P}^2_{1/2}$. For $C \in \texttt{components}(\mathcal{P}_{1/2})$, let $C^1 \in \texttt{components}(\mathcal{P}^1_{1/2})$ denote the subgraph of $\mathcal{P}^1_{1/2}$ isomorphic to C. Similarly, define C^2. Let

$$\triangle E = \operatorname*{argmin}_{\substack{\triangle E' \subseteq V(H_2) \times V(H_2) \\ E(H_1) = E(H_2) \oplus \triangle E'}} |\triangle E'|$$

denote the smallest set of edge modifications to H_2 that produce a graph isomorphic to H_1. We now lower bound $|\triangle E|$. For a component $C \in \mathcal{F}$, let $E(C)$ denote the edge set of that component. Fix a component C such that $E(C) \cap \triangle E \neq \emptyset$. The following cases arise.

- **Case 1:** $\triangle E \cap E(C)$ is connected. Thus, the deletion edits do not disconnect C. In this case, after the insertion edits in $\triangle E$, we obtain a component in $\mathcal{M}_{1/2}$. By the definition of suitable families, if $t > 2/\varepsilon$, it takes at least (follow the sentence). If $t \leq 2/\varepsilon$, then C obviously gets at least one edit, which is at least $\Omega(\varepsilon t)$ edits. In both cases, the number of edits is $\Omega(\varepsilon t)$.

- **Case 2:** $\triangle E \cap E(C)$ is not connected. Now, we get two subcases depending on size of the biggest component in $\triangle E \cap E(C)$.

  1. The biggest component in $C \backslash \triangle E$ has size at least $(1 - 1/100d) \cdot t$. In this case, $\triangle E \oplus E(C)$ maps C to some component in $\mathcal{M}_{1/2}$. Now we split into two cases. Suppose $t > 2/\varepsilon$. Since C is a bounded degree graph, the number of insertion edits (by Definition 3.1) is at least $0.02t - (\frac{0.01}{d} \cdot d)t = 0.01t$. Suppose $t \leq 2/\varepsilon$. In any case C must get at least one insertion edit to make it have size t. So the number of edits is at least $\Omega(\varepsilon t)$.

2. The biggest component in $\triangle E \cap E(C)$ has size at most $(1 - \frac{1}{100d}) \cdot t$. In this case, note that $\triangle E$ removes a balanced separator of C. Thus, the number of deletion edits made to C is at least $\Omega(\varepsilon t)$.

In all of the above cases we see that for any component $C \in \texttt{components}(\texttt{H}_2)$ with $\triangle E \cap E(C) \neq \emptyset$, the number of edits made to C is at least $\Omega(\varepsilon t)$. Finally, note that

$$|\{C \in \texttt{components}(\texttt{H}_2) \colon \texttt{E(C)} \cap \triangle\texttt{E} \neq \emptyset\}| \geq \frac{|\texttt{components}(\texttt{H}_2)|}{2}.$$

And therefore, the distance between graphs H_1 and H_2 is at least $\Omega(\varepsilon)$ as desired. □

We can now define the graph H used to prove the lower bound Theorem 1.2. For convenience, assume that n is a multiple of $t|\mathcal{F}|$, the size of graphs in $\mathcal{F}$. (If not, we will pad the final graph with extra isolated vertices.) The (unlabeled) graph H is a disjoint union of $n/t|\mathcal{F}|$ copies of every graph in $\mathcal{F}$.

Analogously, for $R \subset I, |R| = |\mathcal{F}|/2$, let H_R be the graph that has $n/(2t|\mathcal{F}|)$ copies of every graph in $\mathcal{F}$ indexed by R. For convenience, we refer to any such R as a *half index set*.

3.1 The distinguishing problem

Note that input graph is really a labeled instance, as given by the adjacency list. We now give the explicit distribution of labeled inputs G and the main "distinguishing" problem. By Yao's minimax lemma, it suffices to prove lower bounds for determinstic algorithms over randomized inputs. So let us define the YES and NO distributions.

- YES: Generate a uar labeling of H.

- NO: Generate a uar half index set R. Then, generate a uar labeling of H_R.

Given an input G generated from either of these distribution, the *distinguishing problem* is to determine, with probability $> 2/3$, which distribution G came from. (Note that the supports of these distributions are disjoint.)

Claim 3.4. *If there exists a deterministic distinguisher that makes q queries (where q is independent of n), then there exists an algorithm with the following property. Given input G (generated as above), the algorithm gets q uar connected components of G and determines, with probability $> 2/3 - o(1)$, whether G came from the YES or NO distribution.*

Proof: Consider any deterministic algorithm. One can express its behavior as follows. At any stage, it has explored the adjacency lists of some vertices $v_1, v_2, \ldots, v_b$. It can choose to further query the adjacency lists of these vertices. In this case, it further explores the connected components already visited. The algorithm can also choose to visit a new vertex ID v'. Note that v' is an arbitrary (but fixed) function of all the vertex IDs and edges seen so far. Since the input is a uar labeling of either H or H_R, the probability that v' lies in a component already visited is at most qt/n. With probability at least $1 - qt/n$, v' lies in a uar connected component (since all components have exactly the same size, t). By a union bound, with probability at least $1 - q^2 t/n = 1 - o(1)$, the algorithm sees a uar set of at most q components and makes its decision.

9

The total number of components is n/t, and hence the probability of seeing a specific set of q components is (up to $1 \pm o(1)$-factors) $1/\binom{n/t}{q}$. Since n is sufficiently large, this probability is within $(1 + o(1))$-factors of $q!/(n/t)^q$. This implies that with probability at least $1 - o(1)$, the algorithm sees a uniform distribution of multisets of q components. Alternately, the algorithm gets q uar connected components of G. The success probability only changes by $(1 - o(1))$-factors. $\qquad \square$

The proof of Lemma 3.2 is now a straightforward reduction from distribution testing and an application of the tools built thus far.

Proof: (of Lemma 3.2) By Claim 3.3, all graphs generated by the NO distribution are ε-far from the property $\{H\}$. By Yao's minimax lemma, it suffices to show that any deterministic distinguisher requires $\Omega(\sqrt{|\mathcal{F}|})$ queries.

By Claim 3.4, if there is a deterministic distinguisher making q queries, then there is an algorithm that can distinguish H from H_R (with probability $> 2/3 - o(1)$) by querying q uar connected components. Standard arguments in distribution testing show that $\Omega(\sqrt{|\mathcal{F}|})$ samples are required to distinguish the uniform distribution on the index set $I = [|\mathcal{F}|]$ from the uniform distribution on a uar half index set R (refer to the section titled "An $\Omega(\sqrt{n}/\varepsilon^2)$ lower bound" on page 13 of [Can20]). By mapping I arbitrarily to $\mathcal{F}$, getting samples from the uniform distribution in I is equivalent to getting uar components from G generated from the YES distribution. Analogously, samples from the uniform distribution on a uar half index set R is equivalent to uar components from G generated from the NO distribution. Thus, any algorithm that distinguishes H from H_R with uar connected components requires $\Omega(\sqrt{|\mathcal{F}|})$ samples, leading to the lower bound for deterministic distinguishers. $\qquad \square$

4   A suitable family for planar graphs

The main result of this section is the following theorem.

Theorem 4.1. *There exists a family $\mathcal{F}$ of planar graphs where each graph has s^2 vertices and*
- *$|\mathcal{F}| \geq \exp(\Omega(s^2))$.*
- *For every $G \in \mathcal{F}$, the size of a minimum balanced vertex separator in G is $\Omega(s)$*
- *For every pair of graphs $G, G' \in \mathcal{F}$, it takes at least $0.16s^2$ edge edits to go from G to G'. In particular, it holds that G and G' are 0.02-far.*

We collect some ingredients which will be useful in proving Theorem 4.1. The first important ingredient we need is Whitney's theorem.

Theorem 4.2 (Whitney's Theorem. (Theorem 4.3.2 in [Die17]))**.** *Any two planar embeddings of a 3-connected planar graph are equivalent.*

We first define a "base" graph (Definition 4.3). This is obtained in the following manner. Let us start with the $s \times s$ grid which we denote as G_0. The label set of G_0 is indexed by a pair $(i, j) \in [s] \times [s]$. We call this labeling the *standard grid order*. We denote the base graph by G, and G is constructed on top of G_0 by the addition of some specific edges. Essentially, we add edges at the corners; we refer the reader to Fig. 1. We add edges from $(0,0)$ to $(0,2), (2,0)$, from $(0,s)$ to $(0, s-3), (3, s)$, from (s, s) to $(s, s-4), (s-4, s)$ and from $(s, 0)$ to $(s, 5), (s-5, 0)$. This graph G thus differs from G_0 only in the edges adjacent to the corner vertices, making all the corners

10

unique. Let us label G according to the standard grid order. For $u \in V$, we thus identify u as the pair (x_u, y_u) which refers to the (x, y) coordinate of u on the grid. With this labeling, we have $E(G) = E(G_0) \cup E_1$. where the set E_1 contains all the non grid neighbors of all the corner vertices.

Definition 4.3. *Consider the graph G obtained above and consider the planar embedding of G (which is unique by Theorem 4.2). We will fix an embedding of G where*

*   *The unique corner vertex adjacent to two vertices 2 hops away (according to standard grid order) is located at $(0, 0)$.*
*   *The unique corner vertex adjacent to two vertices 3 hops away (in the standard grid order) is located at $(0, s)$.*
*   *The unique corner vertex adjacent to two vertices 4 hops away (in the standard grid order) is located at (s, s).*
*   *The unique corner vertex adjacent to two vertices 5 hops away (in the standard grid order) is located at $(s, 0)$.*

We call this graph (and by abuse of terminology, its embedding) as the base graph.

Figure 1: An example of one of the graphs from our family $\mathcal{B}$.
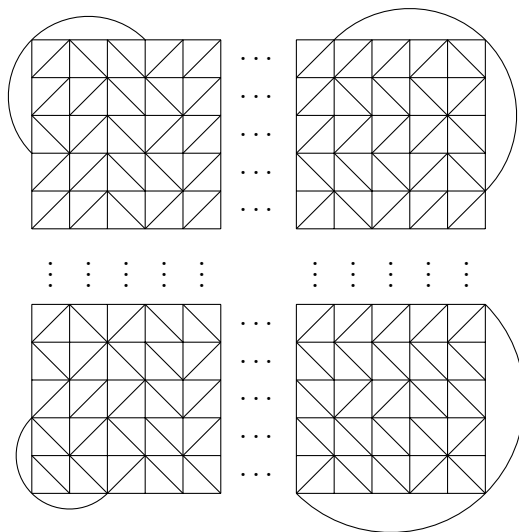
4.1 Proof of Theorem 4.1

Let us begin by defining the following family of graphs. We will show that all graphs in this family are pairwise non isomorphic.

Construction 4.4. *The family $\mathcal{B}$ of graphs is a family of graphs labeled according to* standard grid order *which is obtained from the base graph G in the following way. Let*

$$\mathcal{I} = [0, s) \times (1, s]$$

denote an index set for vertices in $V(G)$. Graphs in this family are obtained in the following manner. For each $(i, j) \in \mathcal{I}$, we add exactly one of the following edges. Either we add

11

- *The upward diagonal $((i, j), (i + 1, j + 1))$, or*

- *The downward diagonal $((i + 1, j), (i + 1, j - 1))$.*

Noting $|\mathcal{I}| = s^2$, note that the size of this family is $|\mathcal{B}| = 2^{s^2}$.

Lemma 4.5. *The family $\mathcal{B}$ defined in Construction 4.4 is a collection of 2^{s^2} pairwise non isomorphic graphs on s^2 vertices.*

Proof: As mentioned earlier in Definition 4.3, the base graph G admits a unique planar drawing. This is due to Theorem 4.2, because the additional edges made our graph 3-connected. We fix that embedding. Now consider two drawings $B_1, B_2 \in \mathcal{B}$. They differ in at least one diagonal and therefore they are non isomorphic. $\square$

However, we still need to show that this family has a sufficient number of graphs that are far from being isomorphic to each other. We state this in the following lemma.

Lemma 4.6. *There is a greedy procedure that takes as input the family $\mathcal{B}$ of graphs and returns a family $\mathcal{F}$ of size at least $2^{s^2/5}$ all of which are pairwise 0.01-far from each other.*

Proof: We begin with the following observations. Consider the ball $B_b(G)$ for any instance of the modified grid G, which is the set of graphs reachable from G by b (diagonal) edge deletions. For any $G \in \mathcal{B}$, note that the number of edges in G is at most s^2. Thus, the number of possible graphs reachable from G after making b deletions is at most $\binom{s^2}{b}$. We replace these deleted diagonals with diagonals oriented the other way. Choose $b = 0.16s^2$. We apply the bound $\binom{a}{b} \leq (ea/b)^b$.

$$\max_{G \in \mathcal{B}} |B_b(G)| \leq \binom{s^2}{b} \leq \left(\frac{es^2}{b}\right)^b \leq \left(\frac{es^2}{0.16s^2}\right)^{0.16s^2} \leq 2^{0.8s^2}$$

Let $\mathcal{F}$ denote the set found by the algorithm below. Using the above bound on $\max_{G \in \mathcal{B}} |B_b(G)|$, it holds that

$$|\mathcal{F}| \geq \frac{2^{s^2}}{\max_{G \in \mathcal{B}} \mathcal{B}_{0.16s^2}(G)} \geq \frac{2^{s^2}}{2^{0.8s^2}} \geq 2^{s^2/5}.$$

Since for any $G \in \mathcal{B}$, the maximum degree of any vertex in G is at most 8, and all the graphs in the set $\mathcal{F}$ disagree in at least $0.16s^2$ edges, it follows that all the graphs in $\mathcal{F}$ are pairwise 0.02-far. Below, we present our greedy procedure.

TakeScoops($\mathcal{B}$)

    1. Initialize $\mathcal{F} = \emptyset$.
    2. **While $\mathcal{B} \neq \emptyset$:**
        (a) $\mathcal{F} = \emptyset$.
        (b) $\mathcal{B} = \mathcal{B} \setminus B_{0.16s^2}(G)$.
        (c) $\mathcal{F} = \mathcal{F} \cup G$.
    end while
    3. **return $\mathcal{F}$.**

The family $\mathcal{F}$ is thus the family output with the desired properties as stated in Theorem 4.1.

$\square$

The proof of the theorem follows immediately from the lemmas.

Proof of Theorem 4.1. We show that the family $\mathcal{F}$ obtained in Lemma 4.6 satisfies all the criteria we require to be satisfied in Theorem 4.1. We pick these one by one:

- $|\mathcal{F}| \geq \exp(\Omega(s^2))$: This is a direct consequence of Lemma 4.6.

- For every $G \in \mathcal{F}$, the size of the minimum balanced separator is $\Omega(s)$: This follows from the fact that the family $\mathcal{F}$ comprises graphs that contain the grid on $s \times s$ vertices as proper induced subgraphs. The grid has balanced separators of size $\Omega(s)$, so all graphs in $\mathcal{F}$ must have separators at least as large.

- Graphs in $\mathcal{F}$ are all pairwise 0.01 far from each other: Follows from Lemma 4.6.

This shows that $\mathcal{F}$ follows the properties described in theorem 4.1, thus completing its proof. $\qquad\square$

4.1.1   Proof of Theorem 1.2

The proof follows from plugging in the right value for s. In Theorem 4.1, we set $s = \varepsilon^{-1}$. We get a suitable family $\mathcal{F}$ with $t = \varepsilon^{-2}$, where $|\mathcal{F}| = \exp(\Omega(\varepsilon^{-2}))$. By Lemma 3.2, we have a explicit singleton property that requires $\Omega(\sqrt{|\mathcal{F}|}) = \exp(\Omega(\varepsilon^{-2}))$ queries to test.

5   A suitable family of trees

This construction is fairly straightforward: we simply pick all distinct trees on ε^{-1} vertices. We need to take care to count the number of *unrooted* trees, so some work is required.

Let $\texttt{UNLAB}_r(s)$ denote the set of unlabled and rooted binary trees on s vertices. We first recall the following fact.

Fact 5.1. *The number of different unlabeled rooted binary trees on s vertices is $|\texttt{UNLAB}_r(s)| \geq 2^{\Omega(s)}$ (Probelm 1.44 in [FS09] where they attribute this result to [Ott48]).*

We would like to lower bound the number of unlabeled, unrooted and binary trees on s vertices. To this end, we first make the following definition.

Definition 5.2. *Write $\texttt{UNLAB}_r(s) = \{(T,r)\colon T \text{ is unlableled with root } r\}$. For $(T,r) \in \texttt{UNLAB}_r(n)$, let*
$$Orbit(T) = \{(T',r') \in \texttt{UNLAB}_r(n) : T' \cong T\}.$$

Note that this also means $T \in Orbit(T')$. Also, note that if $(T'',r'') \notin Orbit(T)$ then $Orbit(T'') \cap Orbit(T) = \emptyset$.

Now, we note the following.

Claim 5.3. *For any $(T,r) \in \texttt{UNLAB}_r(s)$, $|Orbit(T)| \leq s$.*

Proof: Suppose not. Then $\exists (T',x), (T'',x) \in Orbit(T)$ which share x as the root where both $T \cong T'$ and $T \cong T''$. Then $T' \cong T''$ as well. However, we also have $(T',x), (T'',x) \in \texttt{UNLAB}_r(n)$. Contradiction. $\qquad\square$

Now, we will produce a family of unlabeled and unrooted binary trees. This is done by picking a single tree from each orbit in $\mathtt{UNLAB}_r(s)$. The following is immediate.

Claim 5.4. $|\mathit{UNLAB}_u(s)| \geq 2^{\Omega(s)}$.

Proof: Since all orbits have size at most s, it holds that $|\mathtt{UNLAB}_u(s)| \geq |\mathtt{UNLAB}_r(s)| \geq 2^{\Omega(s)}$. $\quad\square$

We can now complete the proof of the main lower bound for forests.

Proof: (of Theorem 1.4) The suitable family is simply $\mathtt{UNLAB}_u(\varepsilon^{-1})$. Since all graphs are of size $t = 1/\varepsilon$, it only suffices to verify the separator bound for suitable families. But that trivially holds, since the separator size is non-zero, which is $\Omega(1) = \Omega(\varepsilon t)$. The size of the family is $\exp(\varepsilon^{-1})$, by Claim 5.4. Lemma 3.2 gives the lower bound of $\exp(\varepsilon^{-1})$ for property testing an explicit singleton property. $\quad\square$

References

[AST94]   Noga Alon, Paul D. Seymour, and Robin Thomas. Planar separators. *SIAM J. Discrete Math.*, 7(2):184–193, 1994. 5

[BGHK95] H.L. Bodlaender, J.R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995. 16

[BSS08]   I. Benjamini, O. Schramm, and A. Shapira. Every minor-closed property of sparse graphs is testable. In *Symposium on the Theory of Computing (STOC)*, pages 393–402, 2008. 2, 4

[Can20]   Clément L. Canonne. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Number 9 in Graduate Surveys. Theory of Computing Library, 2020. 10

[CMOS11] Artur Czumaj, Morteza Monemizadeh, Krzysztof Onak, and Christian Sohler. Planar graphs: Random walks and bipartiteness testing. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*, pages 423–432. IEEE Computer Society, 2011. 2, 4

[CS19]   Artur Czumaj and Christian Sohler. A characterization of graph properties testable for general planar graphs with one-sided error (it's all about forbidden subgraphs). In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1525–1548. IEEE Computer Society, 2019. 2, 4

[Die17]   Reinhard Diestel. *Graph Theory*. Springer Publishing Company, Incorporated, 5th edition, 2017. 10

[EHNO11] Alan Edelman, Avinatan Hassidim, Huy N. Nguyen, and Krzysztof Onak. An efficient partitioning oracle for bounded-treewidth graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques RANDOM*, pages 530–541, 2011. 2, 4

[FS09]   Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. 13

[Gol17]   O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. 4

[Gol21]   Oded Goldreich. Open problems in property testing of graphs. *Electron. Colloquium Comput. Complex.*, page 16, 2021. 2, 4

[GR02]   O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. 1

[Gru12]   Hermann Gruber. On balanced separators, treewidth, and cycle rank. *Journal of Combinatorics*, 3(4):669–681, 2012. 16

[HKNO09]   A. Hassidim, J. Kelner, H. Nguyen, and K. Onak. Local graph partitions for approximation and testing. In *Foundations of Computer Science (FOCS)*, pages 22–31, 2009. 2, 3, 4

[KSS19]   Akash Kumar, C. Seshadhri, and Andrew Stolman. Random walks and forbidden minors II: a poly($d\,\epsilon^{-1}$)-query tester for minor-closed properties of bounded degree graphs. In *STOC 2019.*, pages 559–567, 2019. 2

[KSS21]   Akash Kumar, C. Seshadhri, and Andrew Stolman. Random walks and forbidden minors III: poly($d\,/\epsilon$)-time partition oracles for minor-free graph classes. *Electron. Colloquium Comput. Complex.*, 28:8, 2021. 3, 4, 5

[LR15]   Reut Levi and Dana Ron. A quasi-polynomial time partition oracle for graphs with an excluded minor. *ACM Transactions on Algorithms (TALG)*, 11(3):24, 2015. 2, 3, 4

[LS21]   Reut Levi and Nadav Shoshan. Testing hamiltonicity (and other problems) in minor-free graphs. *CoRR*, abs/2102.11728, 2021. 2, 4

[MR01]   J. Ian Munro and Venkatesh Raman. Succinct representation of balanced parentheses and static trees. *SIAM Journal on Computing*, 31(3):762–776, 2001. 3, 6

[NS13]   Ilan Newman and Christian Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3):1095–1112, 2013. 2, 3, 4, 5

[Ott48]   Richard Otter. The number of trees. *Annals of Mathematics*, 49(3):583–599, 1948. 13

[YI15]   Yuichi Yoshida and Hiro Ito. Testing outerplanarity of bounded degree graphs. *Algorithmica*, 73(1):1–20, 2015. 2

A   Missing proofs from §2

In this appendix, our objective is to prove Claim A.1 and Claim A.4. The former claim counts the number of graphs with treewidth at most τ and is proved below.

Claim A.1. *The number of unlabeled bounded treewidth graphs on t vertices is at most* $\exp O(t \log t)$.

Proof: Observe that for a graph on t vertices with bounded treewidth $\tau = O(1)$ has $O(t)$ edges. Then, the number of unlabelled graphs of this form, T_t is at most $\binom{N}{O(t)}$, where $N = \binom{t}{2}$. Thus $T_t = t^{O}(t)$, which implies that the number of bounded treewidth graphs on t vertices is at most $\exp(O(t \log t))$. □

Next, we produce the statment of Claim A.4. It postulates that there exist efficient $poly(d\varepsilon^{-1})$ time implementations of partition oracles for d-degree bounde graphs with bounded treewidth.

One key tool we need to prove this theorem is an analog of Alon-Seymour-Thomas style result which proves that a graph with treewidth at most τ is $(\varepsilon, O(\tau/\varepsilon))$-hyperfinite.

Let $\tilde{s}(G)$ be the size of the smallest $1/3 - 2/3$ balanced separator, whose removal leads to connected components of size at most $2/3$ of the vertices.

Theorem A.2. *[Theorem 2 of [Gru12] and Theorem 12 of [BGHK95]] $\tilde{s}(G) - 1 \leq tw(G)$.*

We now prove that every treewidth τ graph is $(\varepsilon, 6\tau/\varepsilon)$ hyperfinite.

Lemma A.3. *Fix $\varepsilon > 0$, $d, \tau \in \mathbb{N}$. Then for any graph G with $tw(G) \leq \tau - 1$ there exists a set of edges E' such that the following hold:*

- *$G \setminus E'$ is a graph with connected components no larger than $6\tau/\varepsilon$.*

- *$|E'| \leq \varepsilon dn$*

Proof: We begin by showing item 1. Since G has treewidth at most τ, all of its subgraphs also have treewidth τ. Also, recall $tw(G) \leq \tau$, implies that for all $1/3$-balanced separators of G satisfy $|\texttt{BalSep}(G)| \leq O(\tau)$.

Consider the following recursive procedure. Here $\texttt{Delsep}(G)$ is a routine which returns a three tuple (S, G_0, G_1) where S is the smallest balanced separator of G and G_0 and G_1 are collections of connected components of G each with size at least $|V(G)|/3$.

$\texttt{Decompose}(G)$

    1. If $|V(G)| \geq 6\tau/\varepsilon$
        (a) Write $(S_0, G_0, G_1) = \texttt{Delsep}(G)$.
        (b) $\texttt{Decompose}(G_0), \texttt{Decompose}(G_1)$.

Note that the sets G_0 and G_1 produced at any intermediate step of this procedure are *unions of connected components* (and they are not necessarily connected components themselves). Consider the recursion tree of this process and let us examine the leaves of this recursion tree. The leaf nodes correspond to some subgraph G (which is a union of connected components) obtained after repeated applications of balanced separator routine such that $|V(G)| \leq 6\tau/\varepsilon$. Also, since the procedure repeatedly deletes a $1/3$-separator, it also holds that all the leaf nodes of the recursion tree correspond to graphs with at least $2\tau/\varepsilon$ vertices. Thus, since the leaves correspond to disjoint subgraphs, the number of leaves equals $L \leq \frac{\varepsilon n}{2\tau}$ and the total number of nodes produced in the recursion is at most $2L$. Generating every node requires deleting at most τ edges. Thus, in all, we delete εdn edges which proves the lemma.

□

We now prove bounds on partition oracles for bounded treewidth graphs.

Claim A.4. *Let $\mathcal{T}_\tau$ denote the class of graphs with treewidth at most τ. Let $G \in \mathcal{T}_\tau$ denote a graph with treewidth at most τ. Then G admits an (ε, k) partition oracle with $k \leq \frac{30\tau}{\varepsilon}$.*

Proof: Set $\alpha = \varepsilon/2$. Since $\mathcal{T}_\tau$ is a minor-closed class of graphs, it follows from Theorem 2.3 that any $G \in \mathcal{T}_\tau$ admits an $(\alpha, O(\alpha^{-2}))$ partition oracle whose setting on G is denoted as $\mathcal{P}(G)$. Note that the number of edges that run between the connected components of the underlying partition is at most $\alpha nd \leq 1/2 \cdot \varepsilon dn$. We will show how to modify the parition obtained by this oracle and obtain another oracle which returns components with size at most $30\tau/\varepsilon$. This is done as follows.

Let $P \in \mathcal{P}(G)$ denote some connected component in the partition $\mathcal{P}(G)$. If $|P| \leq 30\tau/\varepsilon$, we do not refine P. On the other hand, if $|P| > 30\tau/\varepsilon$, we use Lemma A.3 on P with parameter $\beta = \varepsilon/2$. This refines P and produces connected components of size at most $3\tau/\beta \leq 6\tau/\varepsilon \leq 30\tau/\varepsilon$ as desired. And the total number of edges lost is at most $\beta dn \leq 1/2 \cdot \varepsilon dn$. Overall, the total number of edges lost is at most εdn as desired. $\square$