Local Graph Partitions for Approximation and Testing

Avinatan Hassidim* MIT avinatan@mit.edu

Jonathan A. Kelner† MIT kelner@mit.edu

Huy N. Nguyen‡ MIT huy2n@mit.edu

Krzysztof Onak§ MIT konak@mit.edu

Abstract—We introduce a new tool for approximation and testing algorithms called *partitioning oracles*. We develop methods for constructing them for any class of bounded-degree graphs with an excluded minor, and in general, for any hyperfinite class of bounded-degree graphs. These oracles utilize only *local* computation to consistently answer queries about a *global* partition that breaks the graph into small connected components by removing only a small fraction of the edges.

We illustrate the power of this technique by using it to extend and simplify a number of previous approximation and testing results for sparse graphs, as well as to provide new results that were unachievable with existing techniques. For instance:

- We give constant-time approximation algorithms for the size of the minimum vertex cover, the minimum dominating set, and the maximum independent set for any class of graphs with an excluded minor.
- We show a simple proof that any minor-closed graph property is testable in constant time in the bounded degree model.
- We prove that it is possible to approximate the distance to almost any hereditary property in any bounded degree hereditary families of graphs. Hereditary properties of interest include bipartiteness, k-colorability, and perfectness.

1. Introduction

Solving combinatorial graph problems (such as minimum vertex cover, maximum independent set, minimum dominating set) has been one of the main research goals of theoretical computer science. In the early 1970s, many of those problems unfortunately turned out to be as hard as the satisfiability problem, due to the breakthrough result of Karp ([12], see the survey [10]). In the 1990s, the discovery of the PCP theorem resulted in showing that even finding good approximate solutions is often NP-hard (see for instance [23]).

In spite of these negative results, multiple methods for finding good approximate solutions for several restricted classes of graphs have been developed over the years. Notably, Lipton and Tarjan [14] proved the separator theorem for planar graphs, which resulted in polynomial-time approximation schemes for several combinatorial problems, which remain NP-hard even restricted to planar graphs [15].

*Supported in part by the Keck Foundation

†Supported in part by NSF Grant CCF-0843915.

‡Supported in part by NSF Award CCF-0832997 and an Akamai Fellowship.

§Supported in part by a Symantec Research Fellowship, NSF Grant 0728645, and NSF Grant 0732334.

The separator theorem was generalized to arbitrary graphs with an excluded minor by Alon, Seymour, and Thomas [1], and similar polynomial-time approximation schemes immediately followed.

An important implication of the separator theorem is that any graph with a fixed excluded minor with maximum degree bounded by d can be partitioned into small components of size at most $\mathrm{poly}(d, 1/\varepsilon)$ by removing only an ε-fraction of edges. In this paper, we develop techniques for locally computing such a partition. We construct a *partitioning oracle* that given query access to a minor-free graph, provides query access to a fixed partition, and uses an amount of computation that is independent of the graph size. Just like knowing the entire partition is useful for finding a good approximate solution, our local version is useful for approximating the size of the optimal solution in time independent of the actual graph size. Our partitioning oracles also find applications to other testing and approximation problems that we describe in more detail below.

Graph classes: We construct partitioning oracles for hyperfinite classes of graphs with bounded degree. Informally, hyperfinite graphs are those that can be partitioned into constant-size components by removing a small fraction of edges. A formal definitions follows.

Definition 1:

- Let $G = (V, E)$ be a graph. G is (ε, k)-*hyperfinite* if it is possible to remove $\varepsilon|V|$ edges of the graph such that the remaining graph has connected components of size at most k.
- Let ρ be a function from $\mathbb{R}_+$ to $\mathbb{R}_+$. A graph G is ρ-*hyperfinite* if for every $\varepsilon > 0$, G is $(\varepsilon, \rho(\varepsilon))$-hyperfinite.
- Let $\mathcal{C}$ be a class of graphs. $\mathcal{C}$ is ρ-*hyperfinite* if every graph in $\mathcal{C}$ is ρ-hyperfinite.

Examples of bounded-degree hyperfinite classes of graphs include bounded-degree graphs with an excluded minor [1] (for instance, bounded-degree planar graphs, bounded-degree graphs with constant tree-width), bounded-degree graphs of subexponential growth [7], and the class of non-expanding bounded-degree graphs considered by Czumaj, Shapira, and Sohler [6].

Elek [8] gives results similar to ours for bounded degree-graphs of subexponential growth. Note that bounded-degree graphs with an excluded minor often do not have bounded growth. For instance, consider full binary trees, which are an example of many popular minor-free classes of graphs. They do not have K_3 as a minor; yet the number of vertices around each vertex grows exponentially fast.

Constant-time approximation algorithms: We say that an algorithm is an *(α, β)-approximation algorithm* for a value $V(x)$ if on input x, it outputs $V'(x)$ such that $V(x) \leq V'(x) \leq \alpha \cdot V(x) + \beta$ with probability at least $2/3$. Moreover, for a given graph problem, we say that an $(\alpha, \varepsilon n)$-approximation algorithm runs in *constant time* if its running time is bounded by a function of ε and the average degree $\tilde{d}$ (which we assume to be known to the algorithm).

The line of research on constant-time approximation algorithms for graph problems was initiated by Parnas and Ron [18], who showed a constant-time $(2, \varepsilon n)$-approximation algorithm for minimum vertex cover[1], and a constant-time $(O(\log d), \varepsilon n)$-approximation algorithm for minimum dominating set. Later, Nguyen and Onak [17] showed a constant-time $(1, \varepsilon n)$-approximation algorithm for maximum matching. Alon[2] showed a constant-time $(O(d \log \log d / \log d), \varepsilon n)$-approximation algorithm for maximum independent set. The running times of some of the above algorithms were improved by Marko and Ron [16], and by Yoshida, Yamamoto, and Ito [22]. On the negative side, Trevisan [18] showed that for any $\delta > 0$, a $(2 - \delta, \varepsilon n)$-approximation algorithm for vertex cover has to make $\Omega(\sqrt{n})$ queries to the graph. Alon showed that there is no constant-time $(o(\log d), \varepsilon n)$-approximation algorithm for minimum dominating set, and no constant-time $(o(d/\log d), \varepsilon n)$-approximation algorithm for maximum independent set.

Elek [8] proved the existence of constant-time $(1, \varepsilon n)$-approximation algorithms for minimum vertex cover, minimum dominating set, and maximum independent set for bounded-degree graphs of subexponential growth. His paper does not provide any explicit bounds on the running time.

In this paper, we show that the above lower bounds can be overcome for any bounded-degree hyperfinite class of graphs. In fact, this is true for a slightly larger family of graph classes with bounded *average* degree, which includes any class of (unbounded degree) graphs with an excluded minor. More precisely, for any such class of graphs, there are constant-time $(1, \varepsilon n)$-approximation algorithms for minimum vertex cover, minimum dominating set, and maximum independent set. For any class of graphs with an excluded minor, the running time of our algorithms is $2^{\mathrm{poly}(1/\varepsilon)}$. Note that finding algorithms of running time $2^{(1/\varepsilon)^{o(1)}}$ is unlikely, since by setting $\varepsilon = 1/(3n)$, this would yield subexponential randomized algorithms for NP-hard problems. The above three problems are NP-hard for planar graphs, even with degree bounded by 3 [9], [10].

[1]From now on, whenever we say "an approximation algorithm for problem A", we mean "an approximation algorithm for the size of the optimal solution to A".

[2]The results of Noga Alon will appear in a joint journal version with [17].

Testing minor-closed properties: Another application of our techniques is to property testing in the bounded-degree model [11]. We say that a graph property[3] is *minor closed* if it is closed under removal of edges, removal of vertices, and edge contraction. Examples of minor-closed families of graphs include planar graphs, outerplanar graphs, graphs of genus bounded by a constant, graphs of tree-width bounded by a constant, and series-parallel graphs.

In the case being considered, the goal of an ε-tester for a given minor-closed property $\mathcal{P}$ is to distinguish, with probability $2/3$, graphs that satisfy $\mathcal{P}$ from those that need to have at least εn edges deleted to satisfy $\mathcal{P}$, where $\varepsilon > 0$. Goldreich and Ron [11] showed an $O(1/\varepsilon^3)$ tester for the property that the input graph is a forest, i.e., does not have K_3 as a minor. Until the breakthrough result of Benjamini, Schramm, and Shapira [4], who showed that any minor-closed property can be tested in constant time, this was the only minor-closed property that was known to be testable in constant time. However, the running time of the tester of Benjamini, Schramm, and Shapira is $2^{2^{2^{\mathrm{poly}(1/\varepsilon)}}}$, and the analysis is quite involved. We give a simple proof of their result, and present a tester that runs in $2^{\mathrm{poly}(1/\varepsilon)}$ time.

Approximation of distance to hereditary properties: A graph property is *hereditary* if it is closed under removal of vertices. Many natural graph families are hereditary, including all minor-closed graph families, perfect graphs, bipartite graphs, k-colorable graphs, graphs with an excluded induced subgraph (see the appendix of [6] for a longer list of hereditary properties). All those properties are known be testable in constant time for dense graphs in the adjacency matrix model even with *one-sided error* [2], i.e., a graph having a given property cannot be rejected. This is not the case in the bounded-degree model. For instance, testing bipartiteness requires $\Omega(\sqrt{n})$ queries [11], and testing three-colorability requires $\Omega(n)$ queries [5]. Motivated by these lower bounds, Czumaj, Shapira, and Sohler [6] turned to testing properties of specific bounded-degree non-expanding families of graphs, which include minor-closed families of graphs. For those graphs, they showed that all hereditary properties are testable in constant-time (with one-sided error). Their proof holds for any hyperfinite family of bounded-degree graphs.

We say that a hereditary property $\mathcal{P}$ is *degenerate* if there is an empty graph on some number of vertices that does not have $\mathcal{P}$. For every non-degenerate hereditary $\mathcal{P}$, every hyperfinite class $\mathcal{C}$ of bounded-degree graphs, and every $\varepsilon > 0$, one can additively approximate the number of edges that must be modified (i.e., inserted or removed) up to εn to achieve $\mathcal{P}$ in time independent of the graph size for graphs in $\mathcal{C}$. It is impossible to specify the running time even for a fixed class of graphs, since $\mathcal{P}$ need not even be computable.

[3]In this paper, all graph properties are defined for graphs with no labels and are therefore closed under permutation of vertices.

Nevertheless, if a non-degenerate $\mathcal{P}$ can be specified via a (potentially infinite) set of forbidden *connected* induced graphs, and there is an $T(n)$-time algorithm that checks if a graph on n vertices has $\mathcal{P}$, then one can show that the distance to $\mathcal{P}$ to can be approximated in any fixed bounded-degree class of graphs with an excluded minor in $2^{\text{poly}(1/\varepsilon)} \cdot T(\text{poly}(1/\varepsilon))$ time.

The reason behind excluding degenerate hereditary properties is the following. Every degenerate $\mathcal{P}$ excludes an empty graph on k vertices, for some constant k, which implies that if a graph G has $\mathcal{P}$, then it does not have an independent set of size k, and therefore, has $\Omega(n^2)$ edges. Since the input graph has $O(n)$ edges, a large number of edges must be inserted. On the contrary, for every non-degenerate hereditary property, the distance to the property is always of order $O(n)$, since it suffices to remove all edges to achieve the property.

A sample application of our result is a $(1, \varepsilon n)$-approximation algorithm for the number of edges that must be removed from the input bounded-degree planar graph to make it 3-colorable. The running time of the algorithm can be made $2^{\text{poly}(1/\varepsilon)}$. The result of Czumaj *et al.* only guarantees the existence of a constant-time algorithm that for planar bounded-degree graphs, can tell 3-colorable graphs from those that need to have at least εn edges removed, for every $\varepsilon > 0$.

Elek [8] proved the existence of constant-time approximation algorithms for distance approximation to union-closed monotone properties in bounded-degree graphs of subexponential growth. Even though a union-closed monotone property need not be hereditary, all natural union-closed monotone properties are hereditary[4]. On the other hand, perfectness is hereditary, but is not monotone.

For general bounded-degree graphs, Marko and Ron [16] give a constant-time $(O(1), \varepsilon)$-approximation algorithm for the distance to H-freeness, where H is an arbitrary fixed graph. They also show a constant-time $(1, \varepsilon)$-approximation algorithm for the distance to cycle-freeness.

Local distributed approximation algorithms: A distributed algorithm is *local* if it runs in a number of rounds that is independent of the size of the underlying graph. The first paper on constant-time graph approximation algorithms due to Parnas and Ron [18] was based on the observation that given a local distributed algorithms for finding a solution to a graph problem, the size of the solution can be approximated by sampling a small number of vertices in the graph, and simulating the distributed algorithm on their neighborhood. On the other hand, Nguyen and Onak [17] noticed that an oracle for a large matching, which they designed for a constant-time algorithm, can be simulated for most nodes on a bounded radius neighborhood. Therefore, a distributed algorithm can construct a good solution in a constant number of communication rounds.

Our partitioning oracle, which provides query access to a partition of vertices, can also be simulated locally. We collect a constant size neighborhood and the random numbers assigned to it. Simulating the oracle's computation for sufficiently many rounds, we assign a partition to most of the nodes, and those that we do not succeed for in the given time limit, create their own parts. This modification cuts additional edges, but their number is small as long as the number of nodes for which simulation does not succeed is small. Such a partition can be used to compute a good approximate solution to many combinatorial problems, in a manner similar to our constant-time algorithms.

Lenzen, Oswald, and Wattenhofer [13] gave a local multiplicative $O(1)$-approximation algorithm for minimum dominating set. Our techniques yield a distributed algorithm that computes an additive $\pm\varepsilon n$ approximation in constant time, for any $\varepsilon > 0$. It is an interesting question if our techniques can be combined with theirs to give a local $(1 + \varepsilon)$-approximation algorithm for this problem.

2. Preliminaries

Model: We assume that an algorithm is given the number n of vertices, and can uniformly sample from the set of vertices in $O(1)$ time. The paper uses the *bounded-degree* model introduced by Goldreich and Ron [11]. In this model, the degrees of all vertices are bounded by $d = O(1)$, and the algorithm has query access to the adjacency list of each vertex. In this paper we assume that we can read a single adjacency list in $O(d)$ time.

Some of our approximation algorithms also work for graphs with bounded *average* degree. We use $\tilde{d}$ to denote a bound on the average degree of a graph. In this case, we assume that an algorithm can learn the degree of a given vertex in $O(1)$ time, and that it can read the adjacency list of a given vertex in time proportional to the number of neighbors.

Partitions: We say that P is a *partition* of a set S if it is a family of nonempty subsets of S such that $\bigcup_{X \in P} X = S$, and for all $X, Y \in P$ either $X = Y$ or $X \cap Y = \emptyset$. We write $P[q]$ to denote the set in P that contains an element $q \in S$.

Uniformity and Non-Uniformity: Throughout the paper, we call a testers, an approximation algorithm, or an oracle *uniform* if it takes ε, the approximation parameter, as input. Otherwise, we call it *non-uniform*.

Graph Minors: A graph H is a *minor* of a graph G, if H can be obtained from G by vertex removals, edges removals, and edge contractions. A graph is H*-minor free* if it does not have H as a minor. A graph property $\mathcal{P}$ is *minor-closed* if for every graph $G \in \mathcal{P}$, every minor of G also belongs to $\mathcal{P}$. The Robertson-Seymour theorem [21] says that every minor-closed property can be expressed via

[4]If a union-closed monotone property is closed under removing an isolated vertex, then it is hereditary. All union-closed monotone properties listed by Elek [8] are hereditary and non-degenerate.

a constant number of excluded minors. Moreover, Robertson and Seymour [20] showed that for every minor H, there is a deterministic $O(n^3)$-time algorithm for checking if H is present in the input graph.

Lemma 2 (Proposition 4.1 in [1]): For every graph H, there exists a constant C_H such that if G is an n-vertex H-minor free graph, then there exists a set S of at most $C_H \cdot n/\sqrt{t}$ vertices of G such that removing vertices of S leaves no connected component on more than t nodes ($t > 1$).

Corollary 3: Let H be a fixed graph. There exists a constant $C_H > 1$ such that for every $\varepsilon \in (0,1)$, every H-minor free graph with degree bounded by d is $(\varepsilon dn, C_H^2/\varepsilon^2)$-hyperfinite.

Proof: Set t in Lemma 2 to $C_H^2/\varepsilon^2 > 1$. One can remove from G at most εn vertices so that each remaining connected component has at most C_H^2/ε^2 vertices. Since the degree of each vertex in G is bounded by d, it suffices to remove from G the edges incident to those vertices to achieve the same property. The number of these edges is at most εdn. ■

Notation: We write $\mathrm{VC}(G)$ to denote the minimum vertex cover size for a graph G.

Let G be a graph. We write $G|_k$, $k \in \mathbb{N}$ to denote G without the edges that are incident to vertices of degree higher than k in G. For a class of graphs $\mathcal{C}$, we define:

$$\mathcal{C}|_k = \{G|_k : G \in \mathcal{C}\}.$$

3. Local Partitions and Their Applications

We now define the main tool that is used in the paper. A partitioning oracle provides query access to a global partition of the graph into.

Definition 4: We say that $\mathcal{O}$ is an *(ε,k)-partitioning oracle* for a class $\mathcal{C}$ of graphs if given query access to a graph $G = (V, E)$ in the adjacency-list model, it provides query access to a partition P of V. For a query about $v \in V$, $\mathcal{O}$ returns $P[v]$. The partition has the following properties:

- P is a function of the graph and random bits of the oracle. In particular, it does not depend on the order of queries to $\mathcal{O}$.
- For every $v \in V$, $|P[v]| \leq k$ and $P[v]$ induces a connected graph in G.
- If G belongs to $\mathcal{C}$, then $|\{(v,w) \in E : P[v] \neq P[w]\}| \leq \varepsilon|V|$ with probability $9/10$.

The most important property of our oracle is that with high probability, it can compute answers in time independent of the graph size by using only local computation. We prove the following lemma for any class of hyperfinite graphs. A relatively simple proof appears in Section 4.

Lemma 5: Let G be an $(\varepsilon, \rho(\varepsilon))$-hyperfinite graph with degree bounded by $d \geq 2$. There is an $(\varepsilon d, \rho(\varepsilon^3/54000))$-partitioning oracle. Let q be the number of non-adaptive queries to the oracle. With probability $1-\delta$, the oracle makes $\frac{q}{\delta} \cdot 2^{d^{O(\rho(\varepsilon^3/54000))}}$ queries to the input graph, and the total amount of the oracle's computation is $\frac{q}{\delta} \log \frac{q}{\delta} \cdot 2^{d^{O(\rho(\varepsilon^3/54000))}}$.

By combining the above oracle with with Corollary 3 one can achieve explicit bounds for any class of graphs with an excluded minor. The expected number of queries of an $(\varepsilon d, C_H/\varepsilon^6)$-partitioning oracle to the input graph is $2^{d^{O(1/\varepsilon^6)}}$ for every query to the oracle. We show a more efficient oracle for bounded-degree ρ-hyperfinite graphs with bounded function ρ.

Lemma 6: Let $R : \mathbb{R}^2 \to \mathbb{R}$ be a polynomial. Let $\mathcal{C}$ be a class of graphs such that, for every $d \in \mathbb{N}_+$, and every $\varepsilon \in (0,1)$, $\mathcal{C}|_d$ is $(\varepsilon, R(d,\varepsilon))$-hyperfinite. There is a uniform partitioning oracle that takes $d \in \mathbb{Z}_+$ and $\varepsilon \in (0,1)$ as input and acts as an $(\varepsilon, \mathrm{poly}(1/\varepsilon, d))$-partitioning oracle for $\mathcal{C}|_d$. Let q be the number of queries to the oracle. The oracle makes $q \cdot 2^{\mathrm{poly}(\varepsilon,d)}$ queries to the input graph and the total amount of the oracle's computation is $(q \log q) \cdot 2^{\mathrm{poly}(\varepsilon,d)}$.

We omit the full proof of the above lemma in this version of the paper. We give a sketch of our techniques in Section 4.

3.1. Constant-Time Approximation Algorithms

We first describe an application of partitioning oracles to approximating the size of an optimal solution for combinatorial problems on restricted classes of graphs. As an example, consider the minimum vertex cover problem for planar graphs. We show that an $(\varepsilon, \mathrm{poly}(1/\varepsilon))$-partitioning oracle for planar graphs of degree $O(1/\varepsilon)$ can be used to partition the input graph into components of size $\mathrm{poly}(1/\varepsilon)$. The union of optimal vertex covers over all connected components constitutes a set of size within $O(\varepsilon n)$ of the minimum vertex cover size of the original graph. By sampling $O(1/\varepsilon^2)$ vertices and checking for each of them if it belongs to the optimal vertex cover for their component, we get a $(1, O(\varepsilon n))$-approximation to the minimum vertex cover size of the original graph.

A formal lemma and proof follow. To achieve a good approximation, a bound on the average degree is needed. Note that every class of graphs with an excluded minor has average degree bounded by a constant.

Lemma 7: Let C be a class of graphs with average degree bounded by $\tilde{d}$. Let $\varepsilon > 0$. Let $\mathcal{O}$ be an $(\varepsilon/3, k)$-partitioning oracle for the class $\mathcal{C}|_{3\tilde{d}/\varepsilon}$. There is a $(1, \varepsilon n)$-approximation algorithm for the minimum vertex cover size in any graph $G = (V, E)$ in $\mathcal{C}$. The algorithm

- gives $\mathcal{O}$ query access to the graph $G|_{3\tilde{d}/\varepsilon}$,
- makes $O(1/\varepsilon^2)$ uniformly distributed queries to $\mathcal{O}$,
- uses $2^{O(k)}/\varepsilon^2 + O(\tilde{d}k/\varepsilon^3)$ time for computation.

The same holds for the maximum independent set problem, and the minimum dominating set problem.

Proof: All edges from G missing in $G|_{3\tilde{d}/\varepsilon}$ can be covered by vertices of degree greater than $3\tilde{d}/\varepsilon$ in G. We write $G' = (V, E')$ to denote $G|_{3\tilde{d}/\varepsilon}$. Note that the number of such vertices is by Markov's inequality at most $\varepsilon n/3$. Therefore, we have

$$\mathrm{VC}(G) - \varepsilon n/3 \leq \mathrm{VC}(G') \leq \mathrm{VC}(G).$$

Each query about the adjacency list of a vertex v in G' can easily be computed in $O(3\tilde{d}/\varepsilon)$ time. If the degree of v is greater than $3\tilde{d}/\varepsilon$ in G, then v is an isolated vertex in G'. Otherwise, we go over the neighbors of v in G, and each neighbor w in G stays a neighbor in G' if and only if w has degree greater than $3\tilde{d}/\varepsilon$ in G. We give $\mathcal{O}$ query access to G'. With probability $9/10$, $\mathcal{O}$ provides query access to a partition P such that the number of edges $(v, w) \in E'$ with $P[v] \neq P[w]$ is at most $\varepsilon n/3$. Let $G'' = (V, E'')$ be G' with those edges removed. Since they can be covered with $\varepsilon n/3$ vertices, we have

$$\mathrm{VC}(G') - \varepsilon n/3 \leq \mathrm{VC}(G'') \leq \mathrm{VC}(G'),$$

that is,

$$\mathrm{VC}(G) - 2\varepsilon n/3 \leq \mathrm{VC}(G'') \leq \mathrm{VC}(G).$$

To get a $(1, \varepsilon n)$-approximation to $\mathrm{VC}(G)$, it suffices to estimate $\mathrm{VC}(G'')$ up to $\pm\varepsilon n/6$. By the Chernoff bound, we achieve that with probability $9/10$ by sampling $O(1/\varepsilon^2)$ vertices and computing the fraction of them in a fixed minimum vertex cover of G''. Such a vertex cover can be obtained by computing a minimum vertex cover for each connected component of G'' independently. Therefore, for every vertex v in the sample, we obtain $P[v]$ from $\mathcal{O}$. We compute a minimum vertex cover for the component induced by $P[v]$ in such a way that the vertex cover does not depend on which vertex in $P[v]$ was the query point. Finally, we check if the query point v belongs to the computed vertex cover for the component. In total, our procedure takes at most $O\left(k \cdot \bar{d}/\varepsilon^3\right) + 2^{O(k)}/\varepsilon^2$ time.

To prove the same statement for maximum independent set, it suffices to notice that removing edges incident to the high degree vertices increases the maximum independent set by at most $\varepsilon n/3$. For minimum dominating set, we assume that all the high degree nodes are in the dominating set, and we take this into account when we compute optimal solutions for each connected component in the partition. This can only increase the solution size by $\varepsilon n/3$. ■

We now use the already recalled fact that the average degree of a graph with an excluded minor is $O(1)$. We combine Lemma 6 and Lemma 7, and achieve the following corollary.

Corollary 8: For every H-minor free family of graphs (with no restriction on the maximum degree), there are $(1, \varepsilon n)$-approximation algorithms for the optimal solution size for minimum vertex cover, maximum independent set, and minimum dominating set that run in $2^{\mathrm{poly}(1/\varepsilon)}$ time.

Algorithm 1: Tester for H-Minor Freeness (for sufficiently large graphs)

```
Input: query access to a partition $P$ given by an
       $(\varepsilon d/4, k)$-partitioning oracle for $H$-minor free
       graphs with degree bounded by $d$ for the input
       graph
1 $f := 0$
2 for $j = 1, \ldots, t_1 = O(1/\varepsilon^2)$ do
3     Pick a random $v \in V$ and a random $i \in [d]$
4     if $v$ has $\geq i$ neighbors, and the $i$-th neighbor of $v$
      not in $P[v]$ then $f := f + 1$
5 if $f/t_1 \geq \frac{3}{8}\varepsilon$ then REJECT
6 Select independently at random a set $S$ of $t_2 = O(1/\varepsilon)$
  vertices of the graph
7 if the graph induced by $\bigcup_{x \in S} P[x]$ is not $H$-minor free
  then REJECT
8 else ACCEPT
```

3.2. Testing Minor-Closed Properties

We now describe how partitioning oracles can be used for testing if a bounded-degree graph has a minor-closed property. The constant-time testability of minor-closed properties was first established by Benjamini, Schramm, and Shapira [4].

We now recall the definition of *property testing* in the bounded degree model [11]. A graph G is *ε-far* from a property $\mathcal{P}$ if it must undergo at least εdn graph operations to satisfy $\mathcal{P}$, where a single graph operation is either an edge removal or an edge insertion. An *ε-tester* T for property $\mathcal{P}$ is a randomized algorithm that has query access to G in the sense defined in the preliminaries, and:

- if G satisfies $\mathcal{P}$, T accepts with probability at least $2/3$,
- if G is ε-far from $\mathcal{P}$, T rejects with probability at least $2/3$.

Lemma 9: Let H be a fixed graph. Let $\mathcal{O}$ be an $(\varepsilon d/4, k)$-partitioning oracle for the class of H-minor free graphs with degree bounded by d. There is an ε-tester for the property of being H-minor free in the bounded-degree model that provides $\mathcal{O}$ with query access to the input graph, makes $O(1/\varepsilon^2)$ uniform queries to $\mathcal{O}$, and uses $O(dk/\varepsilon + k^3/\varepsilon^6) = \mathrm{poly}(d, k, 1/\varepsilon)$ time for computation.

Proof: Our tester is Algorithm 1. The value t_1 equals C_1/ε^2 for a sufficiently high constant C_1 such that by the Chernoff bound the number of edges cut by the partition P is approximated up to $\pm\varepsilon dn/8$ with probability $9/10$. Let $t_3 = C_2/\varepsilon$ be an upper bound on the expected time to hit a set of size $\varepsilon|X|/2$ by independently taking random samples from X, where C_2 is a sufficiently large constant. We set t_2 in the algorithm to $10 \cdot q \cdot t_3$, where q is the number of connected components in H. Finally, we set t_4 to $C_3 \cdot k \cdot t_2^2$ for a sufficiently high constant C_3 such that for graphs on more than t_4 nodes, the probability that two samples from S belong to the same component $P[v]$ is at most $1/10$. If the number of vertices in the graph is at most $t_4 = O(k/\varepsilon^2)$, we read the entire graph, and check if the input is H-minor free in $O((k/\varepsilon^2)^3)$ time. For larger graphs, we run Algorithm 1.

If G is H-minor free, then the fraction of edges cut by P is with probability $1 - 1/10$ at most $\varepsilon dn/4$. If this is the case, the estimate on the number of broken edges that is computed by tester is at most $3\varepsilon dn/8$ with probability $1 - 1/10$. Moreover, every induced subgraph of G is also H-minor free, so G cannot be rejected in the loop in Line 5 of the algorithm. Hence, G is accepted with probability at least $8/10 > 2/3$.

Consider now the case when G is ε-far. If the partition P cuts more than $\varepsilon dn/2$ edges, the graph is rejected with probability $1 - 1/10$. We therefore assume in Steps 6–8 that P cuts less than $\varepsilon dn/2$ edges. Let G' be the new graph after the partition. G' remains $\varepsilon/2$-far from H-minor freeness, and there are at least $\varepsilon dn/2$ edges that must be removed to get an H-minor free graph. This implies that G' is $\varepsilon/2$-far from H_i-minor freeness also for every connected component H_i, $1 \leq i \leq q$, of H. For every i, at least an $\varepsilon/2$-fraction of nodes belong to a component that is not H_i-minor free. Therefore, it suffices to pick in expectation t_3 random nodes to find a component that is not H_i-minor free. For q connected components of H, it suffices to pick in expectation $q \cdot t_3$ random nodes to find each of them. By picking, $10 \cdot q \cdot t_3$ random nodes, we find the components with probability $1 - 1/10$. Furthermore, since the considered graph is large, i.e., has at least t_4 nodes, the components for each i are different with probability $1 - 1/10$, and the graph is rejected in Step 7. Therefore, the probability that a graph that is ε-far is accepted is at most $3/10 < 1/3$. ■

By combining Lemma 6 with Lemma 9, we obtain a $2^{\text{poly}(1/\varepsilon)}$-time tester for H-minor freeness for graphs of degree $O(1)$. Since every minor-closed property can be expressed via a finite set of excluded minors H [21], it suffices to test if the input is ε/s-far from being minor free for each of them, where s is their number. We arrive at the following theorem.

Theorem 10: For every minor-closed property $\mathcal{P}$, there is a uniform ε-tester for $\mathcal{P}$ in the bounded-degree model that runs in $2^{\text{poly}(1/\varepsilon)}$ time.

3.3. Approximating Distance to Hereditary Properties For Hyperfinite Graphs

Parnas, Ron, and Rubinfeld [19] studied generalizations of property testing: *tolerant testing* and *distance approximation*. For a given property $\mathcal{P}$, and an appropriately defined distance to $\mathcal{P}$, an $(\varepsilon_1, \varepsilon_2)$-*tolerant tester* for $\mathcal{P}$ tells apart inputs at distance at most ε_1 from $\mathcal{P}$ and those at distance at least ε_2 from $\mathcal{P}$ with probability at least $2/3$, where $0 \leq \varepsilon_1 < \varepsilon_2$. An (α, β)-*distance approximation algorithm* for $\mathcal{P}$ computes an (α, β)-approximation to the distance of the input to $\mathcal{P}$ with probability $2/3$. In the following, we study constant-time $(1, \delta)$-*distance approximation algorithms* with δ being a parameter. Such algorithms immediately yield constant-time $(\varepsilon_1, \varepsilon_2)$-tolerant testers by setting δ to $(\varepsilon_2 - \varepsilon_1)/2$.

In the bounded-degree model, the *distance* to a given property $\mathcal{P}$ is $k/(dn)$, where k is the minimum number of graph operations (edge insertions and deletions) that are needed to make the graph achieve $\mathcal{P}$. All input graphs have the maximum degree bounded by d, but the closest graph with property $\mathcal{P}$ need not have the degree bounded by d.

Algorithm 2: Approximating distance to not having a set of connected graphs as induced subgraphs

```
Input: set $\mathcal{H}$ of connected graphs (does not include the
       graph on one vertex)
Input: query access to a partition $P$ given by an
       $(\varepsilon d/4, k)$-partitioning oracle for a class $\mathcal{C}$ of
       graphs
1 $f := 0$
2 for $j = 1, \ldots, t = O(1/\varepsilon^2)$ do
3     Pick a random $v \in V$
4     $q :=$ the minimum number of edge operations to
      make the graph induced by $P[v]$ have no graph in
      $\mathcal{H}$ as an induced subgraph
5     $f := f + \frac{q}{d \cdot |P[v]|}$
6 Return $f/t + \varepsilon/2$.
```

Lemma 11: Let $\mathcal{H}$ be a fixed set of connected graphs that does not contain the 1-vertex graph. Let $\mathcal{O}$ be an $(\varepsilon d/4, k)$-partitioning oracle for a class $\mathcal{C}$ of graphs with degree bounded by d, where k is a function of only ε. There is a $(1, \varepsilon)$-approximation algorithm for the distance to the property of not having any graph in $\mathcal{H}$ as an induced subgraph, for graphs in $\mathcal{C}$. The algorithm provides $\mathcal{O}$ with query access to the input graph, makes $O(1/\varepsilon^2)$ random uniformly distributed queries to $\mathcal{O}$, and uses $(O(dk) + 2^{O(k^2)})/\varepsilon^2$ time for computation.

Proof: We use Algorithm 2. The partition P cuts at most $\varepsilon dn/4$ edges with probability $1 - 1/10$, which implies that the distance to the property changes by at most $\pm\varepsilon/4$. Consider the new graph G' with connected components corresponding to the partition of P. Every graph in $H \in \mathcal{H}$ is connected, so H can only appear as an induced subgraph of a connected component of G'. Therefore, it does not make sense to add edges connecting components of G'. This would not exclude any existing induced graph from $\mathcal{H}$. Hence, any shortest sequence of operations that removes from G' all induced copies of graphs in $\mathcal{H}$, does this over each connected component in G' separately.

The value $t = O(1/\varepsilon^2)$ in the algorithm is chosen such that we estimate the number of edge operations divided by

dn up to $\pm\varepsilon/4$ with probability $1 - 1/10$ by the Chernoff bound. Therefore, the algorithm returns a correct estimate with probability at least $1 - 1/10 - 1/10 = 4/5$. The best set of edge operations can be found for a single component in $2^{O(k^2)}$ time by enumerating all possible modifications, and verifying that none of the graphs in $\mathcal{H}$ on at most k nodes are present as an induced subgraph. ■

Lemma 12: Let $\mathcal{P}$ be a non-degenerate hereditary property. Let $\mathcal{O}$ be an $(\varepsilon d/16, k)$-partitioning oracle for a class $\mathcal{C}$ of graphs with degree bounded by d. There is a (non-uniform) $(1, \varepsilon)$-approximation algorithm for the distance to $\mathcal{P}$ for graphs in $\mathcal{C}$. The algorithm provides $\mathcal{O}$ with query access to the input graph, and makes a constant number of uniformly distributed queries to the oracle. Its running time is independent of the graph size.

Proof: The proof reuses some ideas of Czumaj, Shapira, and Sohler [6], who showed a one-sided tester for hereditary properties of hyperfinite classes of bounded-degree graphs.

Let $\mathcal{H}$ be the set of all graphs that do not have $\mathcal{P}$. Since $\mathcal{P}$ is hereditary, if a graph has any of the graphs in $\mathcal{H}$ as an induced subgraph, it does not have $\mathcal{P}$. Consider a subset $\mathcal{H}'$ of $\mathcal{H}$ that only consists of graphs $H \in \mathcal{H}$ that have all components of size at most k. There are at most $t = 2^{O(k^2)}$ different connected graphs $A_1, \ldots, A_t$ on at most k vertices. Every graph in $\mathcal{H}'$ can be represented as a vector $a \in \mathbb{N}^t$, where a_i is the number of times A_i appears as a connected component. For a graph $H \in \mathcal{H}'$, its *configuration* is the vector $c \in \{0,1\}^t$ such that for each i, $1 \leq i \leq t$, $c_i = 0$ if and only if $a_i = 0$. We say that a configuration $c \in \{0,1\}^t$ is *present* if there is a graph in $\mathcal{H}'$ with configuration c. We call the one-vertex graph *trivial*. Recall that $\mathcal{H}$ is non-degenerate. This implies that for each present configuration c, there is i such that $c_i = 1$, and A_i is non-trivial. A subset $\mathcal{X}$ of $\mathcal{A} = \{A_i : 1 \leq i \leq t\}$ is *hitting* if it does not contain the trivial graph, and for every present configuration c, there is j such that $c_j = 1$ and $A_j \in \mathcal{X}$. For non-degenerate $\mathcal{H}$, there always exists at least one hitting subset of $\mathcal{A}$.

Since there exists a $(\varepsilon d/16, k)$-partitioning oracle for the input graph G, G is $(\varepsilon d/16, k)$-hyperfinite, and there is a graph G' with components of size at most k that can be created by removing at most $\varepsilon dn/16$ edges from G. G' is at distance at most $\varepsilon/16$ from G. The distance of G' to $\mathcal{P}$ is bounded from above by the minimum distance from having no induced subgraph in $\mathcal{X}$, where $\mathcal{X}$ is taken over all hitting sets. If we exclude at least one connected component for every graph in $\mathcal{H}'$, we get a graph that satisfies $\mathcal{P}$. We write M to denote the above minimum distance to excluding a hitting set from G'. Note that the shortest sequence of operations that exclude a given hitting set $\mathcal{X}$ does not add edges between different connected components of G. These edges do not remove any existing copy of a graph in $\mathcal{X}$. Note that M is bounded by 1, since it suffices to remove all edges in G' to achieve $\mathcal{P}$.

We now claim that in fact, we have to exclude some hitting set almost entirely for sufficiently large graphs, unless we want to conduct a long sequence of operations. For every present configuration $c \in \{0,1\}^t$ (the number of them is finite), we fix an arbitrary graph $H_c \in \mathcal{H}'$ with this configuration. Consider any sequence of at most $(M - \varepsilon/4) \cdot dn$ operations that turns G' into a graph G''. We will show that for n greater than some constant (which depends on ε, d, k, and $\mathcal{P}$), G'' has an induced copy of one of the graphs H_c. Let $G_\star$ be G'' with only edges that connect vertices in the same connected component in G'. By the definition of M, $G_\star$ must be $\varepsilon/4$-far from having any of the hitting sets excluded. We claim that there is a present configuration c such that for every non-trivial A_i with $c_i = 1$, the distance of $G_\star$ to not having A_i as an induced subgraph is at least $\varepsilon/(8k2^t)$. Suppose for contradiction that for every present configuration c, there is i such that A_i is a non-trivial graph, $c_i = 1$, and the distance of $G_\star$ from not having A_i as an induced subgraph is less than $\varepsilon/(8k2^t)$. For every present configuration c, removing such an A_i from $G_\star$ requires a sequence of fewer than $\varepsilon dn/(8k2^t)$ graph operations. For every inserted or deleted edge (u, v) by such an sequence of operations, let us delete from $G_\star$ all edges incident to both u and v. This is fewer than $\varepsilon dn/(4 \cdot 2^t)$ graph deletions for every present configuration c, and this way we do not introduce any new connected induced subgraph. By going over all present configurations, we can entirely remove all induced copies of at least one graph in each configuration with fewer than $\varepsilon dn/4$ graph deletions. This implies that we can exclude a hitting set with fewer than $\varepsilon dn/4$ graph operations. This yields a contradiction.

We proved that there is a present configuration c such that for every i such that $c_i = 1$ and A_i is non-trivial, the distance of $G_\star$ to not having A_i as an induced subgraph is at least $\varepsilon/(8k2^t)$. Note that because each connected component in $G_\star$ has at most k vertices, the number of vertex disjoint copies of H_c is $\Omega_{\varepsilon,d,k}(n)$ in $G_\star$. Let q be the number of connected components in H_c. We can pick sets I_i, $1 \leq i \leq q$, of subgraphs of $G_\star$ such that each I_i, $1 \leq i \leq q$, is a set of induced copies of the i-th connected component of H_c, $|I_i| \geq \lfloor n/C \rfloor$ (where C only depends on ε, d, k, and the choice of graphs H_c), and the graphs in $\bigcup_i I_i$ are pairwise vertex disjoint. Note that each induced subgraph of $G_\star$ that appears in I_i is also an induced graph in G''. There are at least $\lfloor n/C \rfloor^q$ ways of selecting one subgraph from each I_i. Consider one of such choices. If there were no additional edges between the selected subgraphs, this would give us an induced copy of H_c. The total number of edges in G'' is at most $2dn$, and each edge connects at most 2 subgraphs in $\bigcup I_i$. This means that each edge can make at most n^{q-2} choices of one subgraph from each I_i not give an induced copy of H_c. For sufficiently large n, we have $2dn \cdot n^{q-2} < \lfloor n/C \rfloor^q$, and there is an induced copy of H_c. Summarizing, for sufficiently large graphs, the distance of G' to $\mathcal{P}$ is at least $M - \varepsilon/4$.

Algorithm 3: The global partitioning algorithm with parameters k and δ

```
1  $(\pi_1, \ldots, \pi_n) :=$ random permutation of vertices
2  $P := \emptyset$
3  for $i = 1, \ldots, n$ do
4      if $\pi_i$ still in the graph then
5          if there exists a $(k, \delta)$-isolated neighborhood of $\pi_i$ in the remaining graph then
6              $S :=$ this neighborhood
7          else
8              $S := \{\pi_i\}$
9          $P := P \cup \{S\}$
10         remove vertices in $S$ from the graph
```

Therefore, the distance of G to $\mathcal{P}$ is between $M - 5\varepsilon/16$ and $M + \varepsilon/16$. Moreover, M is approximated up to $\pm\varepsilon/16$ by M', which we define as the distance of G to entirely excluding one of the hitting sets. Therefore, to get a sufficiently good approximation to the distance of G to $\mathcal{P}$, it suffices to compute $(1, \varepsilon n/4)$-approximation to M' for sufficiently large graphs. This can be done by using the algorithm of Lemma 11 for all hitting sets, and amplifying the probability of its success in the standard way. For small graphs, we hard-wire the exact solution to the problem. ∎

4. A Simple Partitioning Oracle

4.1. Local Computation

We reuse a general method for local computation that was introduced by Nguyen and Onak [17]. Consider a graph with random numbers in $[0, 1]$ assigned to its vertices. Suppose that to compute a specific function f of a vertex v, you first need to compute recursively the same function for neighbors of v that were assigned a smaller number than that of v. The following lemma gives a bound on the expected number of vertices for which f must be computed.

Lemma 13 ([17], proof of Lemma 12): Let $G = (V, E)$ be a graph of degree bounded by $D \geq 2$, and let $g : V \times (V \times A)^\star \to A$ be a function. A random number $r(v) \in [0, 1]$ is independently and uniformly assigned to each vertex v of G. A function $f_r : V \to A$ is defined recursively, using g. For each vertex v, we have

$$f_r(v) = g(v, \{(w, f_r(w)) : r(w) < r(v)\}).$$

Let $S \subseteq V$ be a set of vertices v selected independently of r, for which we want to learn $f_r(v)$. The expected number of vertices w for which we have to recursively compute $f_r(w)$ in order to compute $f_r(v)$ for $v \in S$ is at most $|S| \cdot 2^{O(D)}$.

4.2. The Oracle

We introduce an auxiliary definition of a small subset S of vertices that contains a specific node, and has a small number of outgoing edges relatively to S.

Definition 14: Let $G = (V, E)$ be a graph. For any subset $S \subset V$, we write $e_G(S)$ to denote the number of edges in E that have exactly one endpoint in S.

We say that $S \subseteq V$ is a (k, δ)-*isolated neighborhood* of $v \in V$ if $v \in S$, the subgraph induced by S is connected, $|S| \leq k$, and $e_G(S) \leq \delta|S|$.

We now show that a random vertex has an isolated neighborhood of required properties with high probability.

Lemma 15: Let $G = (V, E)$ be a $\rho(\varepsilon)$-hyperfinite graph with degree bounded by d, where $\rho(\varepsilon)$ is a function from $\mathbb{R}_+$ to $\mathbb{R}_+$. Let $G' = (V', E')$ be a subgraph of G that is induced by at least δn vertices. For any $\varepsilon \in (0, 1)$, the probability that a random vertex in G' does not have a $(\rho(\varepsilon^2\delta/1800), \varepsilon/30)$-isolated neighborhood in G' is at most $\varepsilon/30$.

Proof: Any induced subgraph of G can still be partitioned into components of size at most $\rho(\varepsilon)$ by removing at most εn edges. Since G' has at least δn vertices, it is still $(\varepsilon/\delta, \rho(\varepsilon))$-hyperfinite for any $\varepsilon > 0$, or equivalently, it is $(\varepsilon, \rho(\varepsilon \cdot \delta))$-hyperfinite for any $\varepsilon > 0$.

Therefore, there is a set $S' \subseteq E'$ of at most $(\varepsilon^2/1800)|V'|$ edges such that if all the edges in S' are removed, the number of vertices in each connected component is at most $\rho(\varepsilon^2\delta/1800)$. Denote the achieved partition of vertices into connected components by P. We have

$$\mathbb{E}_{v\in V'}\left[\frac{e_G(P[v])}{|P[v]|}\right] = \sum_{S\in P} \frac{|S|}{|V'|} \cdot \frac{e_G(S)}{|S|} = \frac{2|S'|}{|V'|} \leq \frac{\varepsilon^2}{900}.$$

By Markov's inequality, the probability that a random $v \in V'$ is such that $e(P[v])/|P[v]| > \frac{\varepsilon}{30}$ is at most $\varepsilon/30$. Otherwise, $P[v]$ is an $(\rho(\varepsilon^2\delta/1800), \varepsilon/30)$-isolated neighborhood of v. ∎

Finally, we now use the above lemma to construct a partitioning oracle.

Proof of Lemma 5: We set $k = \rho(\varepsilon^3/54000)$ and $\delta = \varepsilon/30$. Consider the global Algorithm 3 with these parameters. The algorithm partitions the vertices of the input graph into sets of size at most k. We define a sequence of random variables X_i, $1 \leq i \leq n$, as follows. X_i corresponds to the i-th vertex removed by Algorithm 3 from the graph. Say, the remaining graph has $n - t$ vertices, and the algorithm is removing a set S of r vertices. Then we set $X_{t+1} = \ldots = X_{t+r} = e_{G'}(S)/r$, where G' is the graph before the removal of S. Note that $\sum_{i=1}^n X_i$ equals the number of edges between different parts in P. For every i, if X_i corresponds to a set S that was a (k, δ)-isolated neighborhood of the sampled vertex, then $X_i \leq \delta = \varepsilon/30$. Otherwise, we only know that $X_i \leq d$. However, by Lemma 15, if $i \leq n - \varepsilon n/30$, this does not happen with probability greater than $\varepsilon/30$. Therefore, we have for every

$i \leq n - \varepsilon n/30$

$$\mathbb{E}[X_i] \leq \varepsilon/30 + d \cdot \varepsilon/30 \leq 2\varepsilon d/30$$

For $i > n - \varepsilon n/30$, we again use the bound $X_i \leq d$. Together, this gives that the expected number of edges connecting different parts of P is at most $2\varepsilon dn/30 + \varepsilon dn/30 < \varepsilon dn/10$. Markov's inequality implies that the number of such edges is at most εdn with probability $9/10$.

It remains to show how Algorithm 3 can be simulated locally. For each vertex v, we want to compute $P[v]$. Instead of a random permutation, we independently assign a random number $r(v)$ uniformly selected from the range $[0, 1]$. We only generate $r(v)$'s when they are necessary, and we store them as they may be needed later again. The numbers generate a random ordering of vertices. To compute $P[v]$, we first recursively compute $P[w]$ for each vertex w with $r(w) < r(v)$ and distance to v at most $2 \cdot k$. If $v \in P[w]$ for one of those w, then $P[v] = P[w]$. Otherwise, we search for a (k, δ)-isolated neighborhood of v, keeping in mind that all vertices in $P[w]$ that we have recursively computed are no longer in the graph. If we find such an neighborhood, we set $P[v]$ to it. Otherwise, we set $P[v] = \{v\}$.

To bound the complexity of the oracle, we use Lemma 13. Our computation graph is $G^\star = (V, E^\star)$ where $E^\star$ connects all pairs of vertices that are at distance at most $2 \cdot k$ in the input graph. The degree of $G^\star$ is bounded by $D = d^{O(\rho(\varepsilon^3/54000))}$. The expected number of vertices for which we have to compute $P[v]$ is at most $q \cdot 2^{d^{O(\rho(\varepsilon^3/54000))}}$. The query complexity at each vertex is bounded by $d^{O(\rho(\varepsilon^3/54000))}$. By Markov's inequality, both the query complexity and the number of vertices visited are bounded by $q \cdot 2^{d^{O(\rho(\varepsilon^3/54000))}}/\delta$ with probability $1 - \delta$.

The required isolated neighborhood of a vertex can easily be found in $2^{d^{O(\rho(\varepsilon^3/54000))}}$ time if it exists. An additional cost in computation comes from the need to find if $r(v)$ was assigned before. By using a standard dictionary, this can be done in time at most logarithmic in the number of $r(v)$ that were assigned. This gives an additional logarithmic factor in the time complexity. ■

5. An Efficient Partitioning Oracle

In this section, we sketch the ideas behind the partitioning oracle of Lemma 6. A detailed description is deferred to the full version of the paper.

5.1. The Partitioning Method

In order to locally simulate Algorithm 3, the simple oracle of Section 4 has to compute the graph partition recursively for a large number of nodes. In particular, it may have to follow long chains of dependencies. In the improved partitioning oracle, we try to avoid such expensive dependencies. Our new global algorithm proceeds in a number of rounds. In each round, it finds a maximal set of disjoint neighborhoods. Then, it removes these neighborhoods from the graph at once and moves on to the next round.

Since each neighborhood removed in round k only depends on the neighborhoods that were removed in previous rounds, its dependency chain has length at most k. If we bound the total number of rounds, then we also bound the number of queries made by each vertex to locally compute its partition. In order to bound the number of rounds, we show that in each round, the expected fraction of vertices removed is at least $\mathrm{poly}(\varepsilon/d)$. Therefore, after a $\mathrm{poly}(d/\varepsilon)$ number of rounds, the remaining graph has less than $\varepsilon dn/2$ edges with high probability, and the algorithm can terminate.

To simulate the algorithm locally, we observe that each round can be simulated for a given vertex v by learning the graph at the end of the previous round within distance $\mathrm{poly}(d/\varepsilon)$ from v. Since there are only $\mathrm{poly}(d/\varepsilon)$ rounds, a vertex can simulate the global algorithm by making at most $d^{\mathrm{poly}(d/\varepsilon)}$ queries to the input graph.

5.2. Growing Neighborhoods

When trying to show that the expected fraction of vertices removed in each round is $\mathrm{poly}(\varepsilon/d)$, it is easy to realize that the naïve neighborhood growing algorithm will not work, since an exponential number of prospective neighborhood can intersect locally. (In such an algorithm, each vertex queries all vertices within distance $\mathrm{poly}(d/\varepsilon)$, and uses brute-force search to find an isolated neighborhood of size $\mathrm{poly}(d/\varepsilon)$.)

Instead, we use the Volume-Biased Evolving Set Process (VBESP) due to Andersen and Peres [3] to grow a neighborhood for each node. With a small modification to the algorithm, we show a lemma similar to their Theorem 2:

Lemma 16: Let $A \subseteq V$ be any $(k, \varepsilon^5/(30^3 \cdot 240 \cdot 960 \cdot d^3 \cdot \log(2kd)))$-isolated neighborhood. There is a subset $A_T \subseteq A$ of size at least $(1-\varepsilon)A$ for which the following holds. If $v \in A_T$, then with probability at least $1 - \varepsilon/60$, we can find an isolated neighborhood S_v will satisfy all the following:

1) S_v is a $(2k, \varepsilon/30)$-isolated neighborhood.
2) $|S_v \setminus A| \leq \varepsilon \cdot |S_v|/30 \cdot d$

The neighborhoods grown by the VBESP have properties that allow for avoiding intersections of too many neighborhoods at the same time. Also, the VBESP runs in near linear time, which is exponentially better than the naïve algorithm. We believe that the VBESP can be helpful for other constant-time algorithms.

6. Open Question

The main open problem is whether there exists a partitioning oracle with query or running time complexity that is polynomial in $1/\varepsilon$ for graphs with an excluded minor. An affirmative answer would imply a $\mathrm{poly}(1/\varepsilon)$ tester for minor-closed properties in the bounded degree model, solving Open Problem 4 in [4].

Acknowledgments

The authors cordially thank Phil Klein for his generosity, his wisdom, and his support throughout the project. Avinatan and Krzysztof thank Erik Demain for an inspiring early discussion. Avinatan thanks Noga Alon for inspiring discussions. Krzysztof thanks Fabian Kuhn for pointing out [13].

References

[1] N. Alon, P. D. Seymour, and R. Thomas, "A separator theorem for graphs with an excluded minor and its applications," in *STOC*, 1990, pp. 293–299.

[2] N. Alon and A. Shapira, "A characterization of the (natural) graph properties testable with one-sided error," *SIAM J. Comput.*, vol. 37, no. 6, pp. 1703–1727, 2008.

[3] R. Andersen and Y. Peres, "Finding sparse cuts locally using evolving sets," *CoRR*, vol. abs/0811.3779, 2008.

[4] I. Benjamini, O. Schramm, and A. Shapira, "Every minor-closed property of sparse graphs is testable," in *STOC*, 2008, pp. 393–402.

[5] A. Bogdanov, K. Obata, and L. Trevisan, "A lower bound for testing 3-colorability in bounded-degree graphs," in *FOCS*, 2002, pp. 93–102.

[6] A. Czumaj, A. Shapira, and C. Sohler, "Testing hereditary properties of nonexpanding bounded-degree graphs," *SIAM J. Comput.*, vol. 38, no. 6, pp. 2499–2510, 2009.

[7] G. Elek, "L^2-spectral invariants and convergent sequences of finite graphs," *Journal of Functional Analysis*, vol. 254, no. 10, pp. 2667 – 2689, 2008.

[8] ——, "Parameter testing with bounded degree graphs of subexponential growth," arXiv:0711.2800v3, 2009.

[9] M. R. Garey and D. S. Johnson, "The rectilinear Steiner tree problem is NP complete," *SIAM Journal of Applied Mathematics*, vol. 32, pp. 826–834, 1977.

[10] ——, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[11] O. Goldreich and D. Ron, "Property testing in bounded degree graphs," *Algorithmica*, vol. 32, no. 2, pp. 302–343, 2002.

[12] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Plenum Press, 1972, pp. 85–103.

[13] C. Lenzen, Y. A. Oswald, and R. Wattenhofer, "What can be approximated locally? Case study: Dominating sets in planar graphs," in *SPAA*, 2008, pp. 46–54.

[14] R. J. Lipton and R. E. Tarjan, "A separator theorem for planar graphs," *SIAM Journal on Applied Mathematics*, vol. 36, pp. 177–189, 1979.

[15] ——, "Applications of a planar separator theorem," *SIAM J. Comput.*, vol. 9, no. 3, pp. 615–627, 1980.

[16] S. Marko and D. Ron, "Approximating the distance to properties in bounded-degree and general sparse graphs," *ACM Transactions on Algorithms*, vol. 5, no. 2, 2009.

[17] H. N. Nguyen and K. Onak, "Constant-time approximation algorithms via local improvements," in *FOCS*, 2008, pp. 327–336.

[18] M. Parnas and D. Ron, "Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms," *Theor. Comput. Sci.*, vol. 381, no. 1-3, pp. 183–196, 2007.

[19] M. Parnas, D. Ron, and R. Rubinfeld, "Tolerant property testing and distance approximation," *J. Comput. Syst. Sci.*, vol. 72, no. 6, pp. 1012–1042, 2006.

[20] N. Robertson and P. D. Seymour, "Graph minors. XIII. The disjoint paths problem," *Journal of Combinatorial Theory, Series B*, vol. 63, no. 1, pp. 65 – 110, 1995.

[21] ——, "Graph minors. XX. Wagner's conjecture," *Journal of Combinatorial Theory, Series B*, vol. 92, no. 2, pp. 325 – 357, 2004, special Issue Dedicated to Professor W.T. Tutte.

[22] Y. Yoshida, M. Yamamoto, and H. Ito, "An improved constant-time approximation algorithm for maximum matchings," in *STOC*, 2009.

[23] D. Zuckerman, "On unapproximable versions of NP-complete problems," *SIAM J. Comput.*, vol. 25, no. 6, pp. 1293–1304, 1996.