Planar Graphs: Random Walks and Bipartiteness Testing

Artur Czumaj*
University of Warwick
Coventry, UK
A.Czumaj@warwick.ac.uk

Morteza Monemizadeh†
University of Frankfurt
Frankfurt, Germany
monemizadeh@em.uni-frankfurt.de

Krzysztof Onak‡
CMU
Pittsburgh, PA, USA
konak@cs.cmu.edu

Christian Sohler§
TU Dortmund
Dortmund, Germany
christian.sohler@tu-dortmund.de

Abstract—We initiate the study of the testability of properties in *arbitrary planar graphs*. We prove that *bipartiteness* can be tested in constant time. The previous bound for this class of graphs was $\tilde{O}(\sqrt{n})$, and the constant-time testability was only known for planar graphs with *bounded degree*. Previously used transformations of unbounded-degree sparse graphs into bounded-degree sparse graphs cannot be used to reduce the problem to the testability of bounded-degree planar graphs. Our approach extends to arbitrary minor-free graphs.

Our algorithm is based on random walks. The challenge here is to analyze random walks for a class of graphs that has good separators, i.e., bad expansion. Standard techniques that use a fast convergence to a uniform distribution do not work in this case. Roughly speaking, our analysis technique self-reduces the problem of finding an odd-length cycle in a multigraph G induced by a collection of cycles to another multigraph G' induced by a set of shorter odd-length cycles, in such a way that when a random walks finds a cycle in G' with probability $p > 0$, then it does so with probability $\lambda(p) > 0$ in G. This reduction is applied until the cycles collapse to self-loops that can be easily detected.

Keywords-property testing, bipartiteness, planar graphs, minor-free graphs, constant-time algorithms

1. Introduction

Property testing studies relaxed decision problems in which one wants to distinguish objects that have a given property from those that are far from this property (see, e.g., [7]). Informally, an object X is ε-far from a property P if one has to modify at least an ε-fraction of X's representation to obtain an object with property P, where ε is typically a small constant. Given oracle access to the input object, a typical property tester achieves this goal by inspecting only a small fraction of the input. Property testing is motivated by the need to understand how to extract information efficiently from massive structured or semi-structured data sets using small random samples.

One of the main and most successful directions in property testing is testing graph properties, which was introduced in papers of Goldreich et al. [8], [9]. There are two popular models, which make different assumptions about how the input graph is represented and how it can be accessed.

In the adjacency matrix model, designed typically for *dense* graphs [8], it is known that testability of a property in constant time is closely related to Szemerédi partitions of the graph. In fact, one can show that a property is testable in constant time, if and only if it can be reduced to testing finitely many Szemerédi partitions [1].

The adjacency list model has been designed mostly for *sparse* graphs. In the most standard scenario, it comes with an additional restriction: the degree of the graph is assumed to be at most a constant d [9]. It is not yet completely understood what graph properties are testable in constant time in this model. Known examples include all hyperfinite properties [16] (see also [3] and [12] for previous general results), connectivity, k-edge-connectivity, the property of being Eulerian [9], and the property of having a perfect matching [17]. On the other hand, some properties testable in constant time in the dense graph model, such as bipartiteness and 3-colorability, are known to require a superconstant number of queries [4], [9].

Even less is known about efficiently testable properties for sparse graphs that do not have a degree bound. It turns out the constant degree bound in the adjacency list model is essential for many of the results mentioned above. All constant-time testers mentioned above use the fact that in a graph with constant maximum degree one can grow some form of a BFS tree of constant depth and then decide based on the obtained information. It is known that connectivity, k-edge-connectivity, and Eulerian graphs are testable in constant time [15]. However, no general results characterizing constant-time testable properties are known.

*Research supported in part by the Centre for Discrete Mathematics and its Applications (DIMAP), EPSRC award EP/D063191/1, by EPSRC award EP/G064679/1, and by a Weizmann-UK Making Connections Grant "The Interplay between Algorithms and Randomness."
†Supported in part by BMBF grant 06FY9097.
‡Supported in part by a Simons Postdoctoral Fellowship and NSF grants 0732334 and 0728645.
§Supported in part by the German Research Foundation (DFG), grant So 514/3-2.

Bipartiteness: The problem of testing bipartiteness has been a great benchmark of the capabilities of property testing algorithms in various graph models. It was one of the first problems studied in detail in both the dense graph model [8] and the sparse graph model [9], [10]. Bipartiteness is known to be testable in $\widetilde{O}(1/\varepsilon^2)$ time in the dense graph model [2], but in the sparse graph model, it requires $\Omega(\sqrt{n})$ queries [9] and is testable in $\tilde{O}(\sqrt{n} \cdot \varepsilon^{-O(1)})$ time [10]. Kaufman et al. [13] show that the property is still testable in $\tilde{O}(\sqrt{n} \cdot \varepsilon^{-O(1)}))$ time in the adjacency list model for graphs that have constant *average* degree.

Czumaj et al. [5] show in the bounded-degree model that if the underlying graph is planar, then any hereditary graph property[1], including bipartiteness, is testable in constant time. This approach can be generalized to any class of graphs that can be partitioned into constant-size components by removing εn edges of the graph, for any $\varepsilon > 0$. Graphs satisfying this property are called *hyperfinite*, and they include all bounded-degree minor-free graphs.

Hassidim et al. [12] show that in fact, the distance to most hereditary properties can be approximated in constant time in such graphs. These results are generalized in the recent work of Newman and Sohler [16], who show that in hyperfinite graphs, one can approximate the distance to *any graph property*. In particular, this implies that *any graph property is testable in hyperfinite graphs*, and therefore, bounded-degree planar graphs.

The central goal of this paper is to initiate the research on the complexity of testing graph properties in general unbounded degree minor-free graphs. Our main technical contribution is the design and analysis of a constant time algorithm testing bipartiteness in arbitrary planar graphs. We show that a constant-length random walk from a random vertex discovers an odd-length cycle in a graph far from bipartite with constant probability. The result easily extends to an arbitrary family of minor-free graphs.

1.1. Techniques

Our approach is based on a new analysis technique for random walks in planar (and minor-free) graphs. We first show that a planar graph that is far from bipartiteness has a linear number of edge-disjoint cycles of constant odd-length. Then we show a contraction procedure that preserves up to a constant factor the probability of discovering an odd-length cycle by a random walk. We show that after a constant number of contractions we obtain a multigraph in which the probability of discovering an odd-length cycle is constant.

1.2. What does not work

Given that bipartitness can be tested in constant time in planar graphs of bounded degree [5], it may seem that there is a simple extension of this result to arbitrary degrees. We now describe simple attempts at reducing our problem to testing bipartiteness in other classes of graphs. We explain why they fail, and we hope this will justify our believe that new techniques are necessary to address the problem.

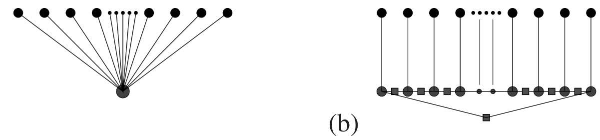
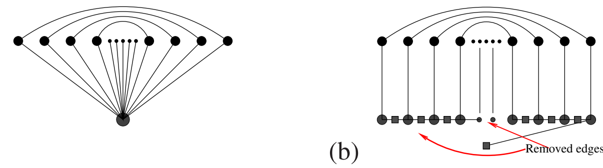

(a) (b)

Figure 1. An example of the process of splitting a vertex that reduces any graph into a graph of maximum degree at most 3 and that maintains planarity. For the graph in (a), figure (b) depicts the splitting that is invariant to being bipartite.



(a) (b)

Figure 2. An example showing that the splitting from Figure 1 can reduce the distance from being bipartite. The planar graph in (a) (in which the ith top vertex from the left is connected by an edge to the ith top vertex from the right) has $\Theta(n)$ edge-disjoint cycles of length 3 and is ε-far from bipartite (one has to remove at least $\frac{n-1}{2}$ edges to obtain a bipartite graph). However, after the splitting, the obtained graph can be made bipartite just by removal of two edges: Figure (b) depicts a bipartite graph obtained after removal of one of the two edges at the bottom and the middle edge in the split part.

Transforming into a constant-degree planar graph: Perhaps the first, and possibly the simplest, approach would be to extend a constant-time algorithm for bounded-degree graphs from [5]. This could be achieved by first transforming an input planar graph G with an arbitrary maximum-degree into a planar graph G^* with bounded-degree and then running the tester for G^* to determine the property for G. However, we do not see any transformation that could work and we do not expect any such transformation to exist.

For example, it is known that one can transform any graph into one with maximum degree at most 3 by splitting every vertex of degree d greater than 3 into d vertices of degree 3. It is also easy to ensure that this reduction maintains the planarity, and also the property of being bipartite (see Figure 1). However, there are two properties that are not maintained: one is the distance from being bipartite (see Figure 2) and another is that the access to the neighboring nodes requires more than a constant time (though this can be "fixed" if one allows each vertex to have its adjacency list ordered consistently with some planar embedding). In particular, Figure 2 depicts an example of a planar graph that is ε-far from bipartite, but after the transformation it suffices to remove 2 edges to obtain a bipartite graph.

Substituting high-degree vertices with expanders: Another transformation of the graph is considered by Kaufman

[1] A graph property is *hereditary* if it is closed under vertex removals.

et al. [13]. They replace every high degree vertex with a constant-degree bipartite expander. While they prove that this construction preserves the distance, it is clear that it cannot preserve the planarity, since planar graphs are not expanders. For expanders, testing bipartiteness can take as much as $\Omega(\sqrt{n})$ queries [9].

2. Preliminaries

Let $G = (V, E)$ be a simple planar graph with $n = |V|$. A graph is *bipartite* if one can partition its vertex set into two sets A and B such that every edge has one endpoint in A and one endpoint in B. We will also frequently use the well known fact that a graph is bipartite if and only if it has no odd-length cycle.

We now formally introduce the notion of being far from bipartiteness[2]. The notion is parameterized by ε.

Definition 1: A planar graph $G = (V, E)$ is *ε-far from bipartite* if one has to delete more than εn edges from G to obtain a bipartite graph.

We are interested in finding a *property testing algorithm* for bipartiteness in planar graphs, i.e., an algorithm that inspects only a very small part of the input graph, and accepts bipartite planar graphs with probability at least $\frac{2}{3}$, and reject planar graphs that are ε-far away from bipartite with probability at least $\frac{2}{3}$, where ε is an additional parameter.

Our algorithm always accepts every bipartite graph. Such a property tester is said to have *one-sided error*.

The access to the graph is given by an *oracle*. We consider the oracle that allows two types of queries:

- *Degree queries:* For every vertex $v \in V$, query the degree of v.
- *Neighbor queries:* For every vertex $v \in V$, query its i-th neighbor.

Observe that by first querying the degree of a vertex, we can always ensure that the i-th neighbor of the vertex exists in the second type of query. In fact, in the algorithm that we describe in this paper, the neighbor query can be replaced with a weaker type of query: *random neighbor query*, which returns a random neighbor of a given vertex v; each time the neighbor is chosen independently, uniformly at random. Finally, we assume that all the above queries take constant time.

In the remainder of the paper we use several constants depending on ε. We use lower case Greek letters to denote constants that are typically smaller than 1 (e.g., $\delta_i(\varepsilon)$) and lower case Latin letters to denote constants that are usually larger than 1 (e.g., $f_i(\varepsilon)$). All these constants are always positive.

[2]The standard definition of being ε-far (see for example the definition in [13]) expresses the distance as the fraction of edges that must be modified in G to obtain a bipartite graph. Compared to our Definition 1, instead of deleting εn edges, one can delete εm edges, where m is the number of edges. For any class of graphs with an excluded minor, the number of edges in the graph is upper bounded by $C \cdot n$, where C is a constant. Moreover, unless the graph is very sparse (i.e., most of its vertices are isolated, in which case even finding a single edge in the graph may take a large amount of time), the number of edges in the graph is at least $\Omega(n)$. Thus, under the standard assumption that $m = \Omega(n)$, the ε in our definition and the ε in the previous definitions remain within a constant factor. We use our definition of being far for simplicity.

2.1. Basic lemmas

We begin with a lemma that trivially follows from the Klein-Plotkin-Rao decomposition theorem [14].

Lemma 2: Let $G = (V, E)$ be a simple planar graph and let δ be a parameter in $(0, 1)$. There is a set of at most $\delta|E|$ edges in G whose deletion decomposes G into connected components where the distance (in the original graph) between any two nodes in the same component is $O(1/\delta^2)$.

Lemma 2 leads to the following key property of planar graphs that are ε-far from bipartite.

Lemma 3: Let G be a simple planar graph. If G is ε-far from bipartite, then G has at least $\frac{\varepsilon n}{q(\varepsilon)}$ edge-disjoint odd-length cycles of length at most $\frac{1}{2}q(\varepsilon)$ each, where $q(\varepsilon) = O(1/\varepsilon^2)$.

Proof: We define $q(\varepsilon)$ such that $q(\varepsilon) \geq 6$ and $\frac{1}{6}q(\varepsilon)$ is a bound on the diameter (in the original graph) of components in the decomposition given by Lemma 2 with $\delta = \frac{1}{2}\varepsilon$.

We find the cycles one by one. Suppose that we have already found in G a set of k edge-disjoint odd-length cycles of length at most $\frac{1}{2}q(\varepsilon)$ each, where $k < \frac{\varepsilon n}{q(\varepsilon)}$. We show the existence of one more such cycle, which by induction yields the lemma.

Let G^* be the subgraph of G obtained by removing the k edge-disjoint odd-length cycles of length at most $\frac{1}{2}q(\varepsilon)$ each. Since $k < \frac{\varepsilon n}{q(\varepsilon)}$, G^* is $\varepsilon/2$-far from bipartite. Apply Lemma 2 to G^* with $\delta = \frac{1}{2}\varepsilon$ and let H be the resulting decomposition. Since G^* is $\varepsilon/2$-far from bipartite, H is not bipartite. Let us consider a connected component C_H of H that is not bipartite and let v be a vertex from C_H. Build a BFS tree from v in G^*. Since C_H is not bipartite, there must be two vertices u_1 and u_2 in C_H that have the same distance from v and that are connected by an edge in H (otherwise, we could define a bipartition of C_H by the parity of the distance from v in the BFS tree). Let v' be the last common vertex on the paths from v to u_1 and from v to u_2 in the BFS tree. The tour that starts at v', goes to u_1 via the BFS tree edges, then takes the edge connecting u_1 and u_2, and finally returns to v' via the BFS tree edges is a odd-length cycle of length at most $\frac{1}{3}q(\varepsilon) + 1 \leq \frac{1}{2}q(\varepsilon)$. ■

3. Algorithm Random Bipartitness Exploration

We now give our algorithm for testing bipartiteness of planar graphs with arbitrary degree and provide an overview of its analysis. A detailed proof appears Sections 4 and 5.

Random Bipartiteness Exploration $(\mathbf{G}, \varepsilon)$**:**

- Repeat $f(\varepsilon)$ times:
  - Pick a random vertex $v \in V$
  - Simulate a random walk of length $g(\varepsilon)$ from v
  - If the random walk found an odd-length cycle, then **reject**
- If none of the random walks found an odd-length cycle, then **accept**

Theorem 4: There are functions f and g such that

- if G is bipartite then Random Bipartiteness Exploration (G, ε) accepts G, and
- if G is ε-far from bipartite then Random Bipartitness Exploration (G, ε) rejects G with probability at least 0.99.

Let us first observe that the first claim is obvious: if G is bipartite, then every subgraph of G is bipartite as well, and Random Bipartitness Exploration always accepts. Thus, to prove Theorem 4, it suffices to show that if G is ε-far from bipartite, then Random Bipartitness Exploration rejects G with probability at least 0.99.

Therefore, from now on, we assume that the input graph G is ε-far from bipartite. Then by Lemma 3, we know that G has at least $\frac{\varepsilon n}{2q(\varepsilon)}$ edge-disjoint odd-length cycles of length at most $q(\varepsilon)$ each, where $q(\varepsilon) = O(1/\varepsilon^2)$. Let us denote by $\mathcal{C}^\diamond$ any such set of edge-disjoint odd-length cycles. Thus, to complete the proof, it suffices to show that Random Bipartitness Exploration finds *one* of the odd-length cycles from $\mathcal{C}^\diamond$ (with probability at least 0.99).

Unfortunately, this approach applied directly cannot work, as one can see in Figure 3. Instead, we prove a sufficient result that Random Bipartitness Exploration finds a short odd-length cycle that is a *combination of the cycles from* $\mathcal{C}^\diamond$ (with probability at least 0.99).

Finally, note that it suffices to show that a *single* random walk of length $g(\varepsilon)$ finds an odd-length cycle with probability at least $5/f(\varepsilon)$. This immediately implies that $f(\varepsilon)$ independent random walks detect at least one odd-length cycle with probability at least $1-(1-5/f(\varepsilon))^{f(\varepsilon)} \geq 1-e^{-5} \geq 0.99$.

3.1. Overview of the analysis

The proof of Theorem 4 is split into two sections. First, in Section 4, we reduce the problem, along the lines described above, to that of finding an odd-length cycle in a subgraph of the original graph G that is induced by a linear number of edge-disjoint odd-length cycles of constant length. Once we have this reduction, we next consider weighted graphs and multigraphs induced by odd-length cycles. These multigraphs result from the original set by cycle removals and certain edge contractions. A vertex with weight w in such a multigraph stands for a set of w vertices in the original graph, and weighted sampling from the vertices is equivalent to uniform sampling in the original graph. Thus, each multigraph represents a possibly contracted set of cycles from the original graph. We show in Section 5 that for any such multigraph induced by a linear number of odd-length cycles, we can remove a constant fraction of cycles such that every remaining cycle can be contracted without interfering with other cycles. This reduces the length of these cycles by one. Furthermore, if a random walk finds a cycle in the remaining cycles with probability p, then it does so with probability $\eta(p)$ in the original set. Thus, if we apply this technique until all cycles are self-loops, we can easily prove that a random walk finds such a self-loop and our reduction ensures that this happens with constant probability for the original graph.

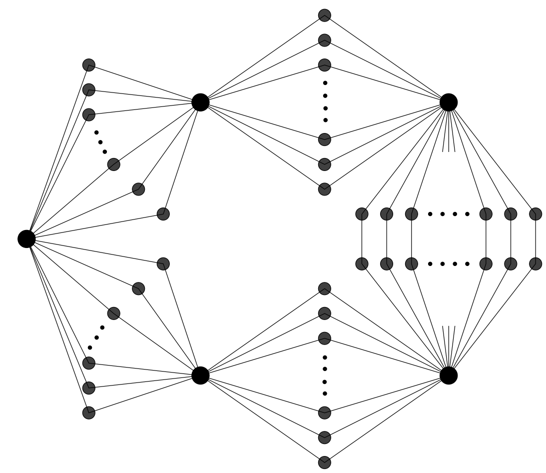


Figure 3. A planar graph G with $\frac{n-5}{6}$ edge-disjoint odd-length cycles. Each of the five high degree vertices has degree exactly $\frac{2(n-5)}{6}$, and the edge-disjoint cycles are of length 11 each. Observe that if $\mathcal{C}^\diamond$ is any fixed set of $\frac{n-5}{6}$ edge-disjoint cycles of length 11, then the probability that a single constant-length random walk discovers one of the cycles in $\mathcal{C}^\diamond$ is polynomially small. Nevertheless, a single random walk of length, say, 12 finds *an* odd-length cycle with probability at least 2^{-11}.

4. First reduction: Dealing with graphs induced by odd-length cycles

Let $\mathcal{C}$ be a set of cycles on a vertex set V. We denote by $\mathcal{G}(\mathcal{C})$ the graph induced by $\mathcal{C}$. That is, $\mathcal{G}(\mathcal{C}) = (V, E_{\mathcal{C}})$ with $E_{\mathcal{C}}$ being the set of the edges from the cycles in $\mathcal{C}$.

Our first lemma states that in order to show that an l-step random walk from a random start vertex finds an odd-length cycle in a planar graph G that is ε-far from bipartite, it suffices to show that the random walk finds an odd-length cycle in a subgraph of G induced by a linear number of edge-disjoint odd-length cycles of constant length.

Lemma 5: Let $G = (V, E)$ be a simple planar graph that has a set $\mathcal{C}^*$ of at least αn edge-disjoint odd-length cycles in G and let $l > 0$ be an integer. There there is $\zeta = \zeta(l, \alpha) > 0$ and there is a subset $\mathcal{C}$ of $\mathcal{C}^*$ of size at least $\frac{1}{2}\alpha n$ such that

- if the probability that an l-step random walk in $\mathcal{G}(\mathcal{C})$ starting from a random vertex finds an odd-length cycle in $\mathcal{G}(\mathcal{C})$ is p, then
- the probability that an l-step random walk starting from a random vertex finds an odd-length cycle in G is at least $\zeta \cdot p$.

Proof: To construct the subset $\mathcal{C}$, we first delete some cycles from $\mathcal{C}^*$. The process of deleting the cycles is based on the comparison of the original degree of the vertices with the current degree in $\mathcal{G}(\mathcal{C}^*)$. To implement this scheme, we write $\deg_G(v)$ to denote the degree of v in the original graph G and we use the term *current degree* of a vertex v to denote its current degree in the graph $\mathcal{G}(\mathcal{C}^*)$ induced by the *current* set of cycles $\mathcal{C}^*$.

We repeat the following procedure as long as possible: if there is a vertex $v \in V$ with current degree in $\mathcal{G}(\mathcal{C}^*)$ at most $\frac{1}{12}\alpha \deg_G(v)$, then we delete from $\mathcal{C}^*$ all cycles going through v in $\mathcal{C}^*$. To estimate the number of cycles deleted, we charge to v the number of deleted cycles in each such operation.

Let $\mathcal{C}$ be the remaining set of cycles from $\mathcal{C}^*$. Observe that each $v \in V$ can be processed above at most once. Indeed, once v has been used, it becomes isolated and hence it is not used again. Therefore, at most $\frac{1}{12}\alpha \deg_G(v)$ cycles from $\mathcal{C}$ can be charged to any single vertex. This, together with the inequality $\sum_{v\in V} \deg_G(v) \le 6n$ by planarity of $\mathcal{G}(\mathcal{C}^*)$, implies that the total number of cycles removed from $\mathcal{C}^*$ is upper bounded by $\sum_{v\in V} \frac{1}{12}\alpha \deg_G(v) \le \frac{1}{2}\alpha n$. Since $|\mathcal{C}^*| \ge \alpha n$, we conclude that $|\mathcal{C}| \ge |\mathcal{C}^*| - \frac{1}{2}\alpha n \ge \frac{1}{2}\alpha n$.

We have constructed a subset $\mathcal{C}$ of $\mathcal{C}^*$ of size at least $\frac{1}{2}\alpha n$ such that for every vertex $v \in V$, either v is isolated in $\mathcal{G}(\mathcal{C})$ or its degree is greater than $\frac{1}{12}\alpha \deg_G(v)$ (i.e., at least $\frac{\alpha}{12}$ fraction of its original degree in G). We now use this property to show that if the probability that an l-step random walk starting from a random vertex in $\mathcal{G}(\mathcal{C})$ finds an odd-length cycle is p, then an l-step random walk starting from a random vertex in G finds an odd-length cycle with probability at least $p\cdot\zeta$, for appropriately chosen $\zeta = \zeta(l, \alpha)$. This will yield the theorem.

Let us consider a fixed sequence of $l+1$ vertices $\langle x_0, x_1, \dots, x_l\rangle$ in $\mathcal{G}(\mathcal{C}^*)$ with $(x_i, x_{i+1}) \in E_{\mathcal{C}}$, $0 \le i \le l-1$, that contains an odd-length cycle $\mathfrak{c}$. Our claim is that, if the probability of this fixed sequence to be chosen as an l-step random walk in $\mathcal{G}(\mathcal{C})$ is p' then it is at least $\zeta \cdot p'$ in G.

Since x_0 is one of the starting vertices, and since x_0 cannot be isolated in $\mathcal{G}(\mathcal{C})$ (because $\mathcal{G}(\mathcal{C})$ has edge (x_0, x_1)), we must have $\deg_{\mathcal{G}(\mathcal{C})}(x_0) > \frac{\alpha}{12}\deg_G(x_0)$. Therefore, when the random walk in G chooses a neighbor of x_0, it chooses x_1 with probability at least $\frac{\alpha}{12}$ times the probability that the random walk in $\mathcal{G}(\mathcal{C})$ chooses x_1. The same arguments can be used to argue that if the random walk in G has chosen any vertex x_i, then it also chooses x_{i+1} with the probability at least $\frac{\alpha}{12}$ times the respective probability for the random walk in $\mathcal{G}(\mathcal{C})$. Therefore, if the random walk in $\mathcal{G}(\mathcal{C})$ is chosen with probability p' then the probability of choosing the same random walk in G is at least $(\frac{\alpha}{12})^l \cdot p'$. Summing up over all sequences of $l+1$ vertices $\langle x_0, x_1, \dots, x_l\rangle$ in $\mathcal{G}(\mathcal{C})$ with $(x_i, x_{i+1}) \in E_{\mathcal{C}}$, $0 \le i \le l-1$, and that contain an odd length cycle, we obtain the claim with $\zeta = (\frac{\alpha}{12})^l$. ■

Due to Lemma 5, we now turn our attention only to graphs and multigraphs induced by odd-length cycles.

5. Second reduction: Analysis for graphs and multigraphs induced by odd-length cycles

We continue with the assumption that the input graph G is planar and ε-far from bipartite. By Lemma 3 we know that this graph contains at least $\varepsilon n/q(\varepsilon)$ cycles of length $\ell(\varepsilon) := q(\varepsilon)/2$ for a $q(\varepsilon) = O(1/\varepsilon^2)$. Let $\mathcal{C}^*$ be such a set of cycles. We use the set $\mathcal{C}^*$ and apply Lemma 5. Lemma 5 states that, in order to show that a random walk finds (with constant probability) an odd-length cycle in G, it is enough to show that a random walk finds (with constant probability) an odd-length cycle in the graph $G(\mathcal{C})$ induced by a set $\mathcal{C} \subseteq \mathcal{C}^*$ of $\alpha(\varepsilon)n$ edge-disjoint odd-length cycles in G, $\alpha(\varepsilon) := \varepsilon/(2q(\varepsilon))$. Furthermore, by our choice of $\mathcal{C}^*$ each cycle in $\mathcal{C}$ has length at most $\ell(\varepsilon)$. Thus, in the following we assume that a set of cycles $\mathcal{C}$ with such properties is given and we show that with constant probability a random walk finds a cycle in the graph $G(\mathcal{C})$. By choosing an appropriate function f in algorithm Random Bipartiteness Exploration, this constant can be made arbitrarily close to one.

To show that a random walk finds a cycle with constant probability, we use a second reduction. We now give some intuition for this second reduction.

5.1. Overview

We first observe that, if all vertex degrees in $G(\mathcal{C})$ are constant, then a single random walk finds a cycle with constant probability. This is because with constant probability the starting vertex is on some cycle C of $\mathcal{C}$ and in each step we follow C with constant probability. Since C is a cycle of length at most $\ell(\varepsilon)$, the observation follows. Thus, the hard part is to deal with vertices whose degree is not constant. An example that captures many of the difficulties of the problem is given in Figure 3. In this example, we have many parallel cycles that intersect at several vertices of high degree. However, many vertices of the cycles also have constant degree. By the linear bound on the number of edges in a planar graph, this is the case for any planar graph.

Our main idea is now to contract paths of length 2 whose middle vertex has constant degree to a single edge. The motivation behind this contraction is that if a random walk takes the first edge of such a path, then with constant probability it also follows the second edge. Thus, from the point of the analysis of the random walk, we can view this path as a single edge. Furthermore, since many cycles must have at least one subpath whose middle vertex has constant degree, we can make sure that a constant fraction of cycles has at least one subpath that is contracted. If we only keep the cycles of $\mathcal{C}$ that have been contracted, we end up with a linear-size set $\mathcal{C}'$ of cycles with length reduced by at least 1. Furthermore, since planarity is closed under edge contraction and edge removal, the graph $G(\mathcal{C}')$ is planar again. This allows us to repeatedly apply our reduction until all cycles collapse to self-loops. Such a repeated application may be necessary since $G(\mathcal{C}')$ may still have vertices of high degree. Then since a constant fraction of edges belongs to self-loops, a random walk traverses a self-loop with constant probability and so, by the properties of our reductions, with constant probability a random walks finds a cycle in the original graph.

In the following, we develop a framework that formalizes our idea. In particular, we need to deal with the following technical problems.

- The starting vertex of a random walk in $G = (V, E)$ is chosen uniformly at random from V. Edge contractions change this probability. We use *vertex weights* to keep track of this change in probability.
- We are interested in finding cycles of odd length. Thus, if we contract a path of length 2 to a single edge, this changes the parity of all cycles that contain this path from odd to even or vice versa. In order to keep track of these changes we assign *parities* to edges (at the beginning all parities are odd). If we contract two odd edges or two even edges, the resulting edge is even. Otherwise, it is odd.
- We need to remove parallel edges to exploit planarity in the contracted graphs. For this purpose, we introduce edge weights such that the *weight of an edge* (v, u) can be interpreted as the number of parallel edges between v and u.

5.2. The framework

We begin with a set of definitions and concepts used later in our analysis.

In this section we consider *multigraphs*, i.e., graphs with *parallel edges*. We also allow *self-loops*. We sometimes represent a multigraph as an *edge-weighted graph*, whose edge weights correspond to the multiplicity of the edges (edge-weighted graphs may have self-loops). We also consider weighted graphs whose vertices and edges have weights.

We extend our definition of graphs induced by cycles (Section 4) to multigraphs. Let $\mathcal{C}$ be any *multiset* of cycles on vertex set V. We allow the cycles in $\mathcal{C}$ to share edges. We denote by $\mathcal{G}(\mathcal{C})$ the multigraph induced by $\mathcal{C}$, that is, $\mathcal{G}(\mathcal{C}) = (V, E_{\mathcal{C}})$ is a multigraph with $E_{\mathcal{C}}$ being the set of the edges on the cycles in $\mathcal{C}$; if an edge e appears in multiple cycles in $\mathcal{C}$ then the same number of copies of e appears in $E_{\mathcal{C}}$.

For the study of bipartiteness, we consider labeled graphs or multigraphs, where each edge of a graph or a multigraph is labeled either *odd* or *even*. The intuition is that if an edge has label odd (or even), then this edge corresponds to an odd-length (or even-length, respectively) path (or a cycle, in the case when the edge is a self-loop) in the original graph. We also define an operation of XOR on a set of edge labels $\mathcal{L}$: if the number of labels *odd* in $\mathcal{L}$ is even then the XOR returns label *even*; otherwise, it returns label *odd*. We call a path or a cycle *even* if the number of labels *odd* on its edges is even; it is called *odd* otherwise.

A pair of vertices $\langle x, y\rangle$ is called *τ-parity balanced for $\mathcal{C}$* if either all parallel edges (x, y) in $\mathcal{G}(\mathcal{C})$ have the same parity (that is, all are odd, or all are even), or the ratio between *odd* parallel edges (x, y) and *even* parallel edges (x, y) in $\mathcal{G}(\mathcal{C})$ is at least τ and at most $\frac{1}{\tau}$.

Random walks on multigraphs: A random walk in a multigraph selects at each vertex every edge incident to this vertex with the same probability. The probability that a single step of a random walk moves from v to u equals the number of edges between v and u divided by the total number of edges (with multiplicities) that are incident to v.

5.2.1. Contractions of cycles, cycle-minors, and off-springs: In our analysis, we apply *a sequence of contractions* to the graph/multigraph induced by a set of cycles. This operation contracts some paths to edges and simplifies the structure of the graph, as needed in our analysis. We begin with definitions used.

If $\mathcal{C}$ is a set of cycles on V, then for every $v \in V$, we denote by $\mathcal{C}_v$ the set of cycles in $\mathcal{C}$ going through v.

Loop vertices: Let $\mathcal{C}$ be any set of cycles on a vertex set V. We say a *vertex $v \in V$ is a loop vertex* (with respect to $\mathcal{C}$) if $\mathcal{C}_v \neq \emptyset$ and every cycle in $\mathcal{C}_v$ is a self-loop at v.

Contractible vertices: Let $\mathcal{C}$ be any set of cycles on a vertex set V. We say a *vertex $v \in V$ is contractible* (with respect to $\mathcal{C}$) if $\mathcal{C}_v \neq \emptyset$, and there are two vertices $x, y \in V$ (it is allowed that $x = y$), such that every cycle in $\mathcal{C}_v$ enters v through vertex x and leaves v through vertex y.

Cycle contraction at a vertex: Let $\mathcal{C}$ be any set of cycles on a weighted vertex set V. Let v be a contractible vertex with respect to $\mathcal{C}$. Let x, y be two vertices in V, such that every cycle in $\mathcal{C}_v$ enters v through vertex x and leaves v through vertex y. Modify every cycle $\mathfrak{c} \in \mathcal{C}_v$ by contracting path $\langle x, v, y\rangle$ into edge (x, y), such that the label of the new edge is the XOR of the labels of edges (x, v) and (v, y) from $\mathfrak{c}$. We call the path $\langle x, v, y\rangle$ the *offspring* of the obtained new edge (x, y). Let $\mathcal{C}^*_v$ be the resulting set of cycles. Then for a set $\mathcal{C}$ of cycles on a weighted vertex set V, *cycle contraction* at a vertex $v \in V$ is the operation of replacing $\mathcal{C}$ by $\mathcal{C} \setminus \mathcal{C}_v \cup \mathcal{C}^*_v$, and changing the weight of vertices $v, x, y \in V$ by distributing the weight of v equally to vertices x and y, and then zeroing the weight of v. (In particular, if $x = y$ (i.e., (x, y) is a self-loop) then the weight of x increases by the weight of v.)

Definition 6 (Cycle-minor): Let $\mathcal{C}$ be a set of labeled cycles on a weighted vertex set V. Any set $\mathcal{C}'$ of labeled cycles on a weighted vertex set V obtained from $\mathcal{C}$ by applying an arbitrary sequence of cycle removals and cycle contractions at a vertex is called a *cycle-minor* of $\mathcal{C}$.

The following observation will be crucial for our second reduction. It follows from the fact that planarity is closed under contraction of edges and removal of edges (and vertices).

Observation 7: If $\mathcal{C}$ is a collection of cycles such that $\mathcal{G}(\mathcal{C})$ is planar, then for any cycle minor $\mathcal{C}'$ of $\mathcal{C}$, the multigraph $\mathcal{G}(\mathcal{C}')$ represented as an edge weighted graph is planar.

Let us state some further useful properties of vertex weights in cycle-minors.

Lemma 8 (Vertex-weight Lemma): Let $\mathcal{C}$ be an edge-disjoint set of cycles on a weighted vertex set V with weight$(v) = 1$ for every $v \in V$ and such that $\mathcal{G}(\mathcal{C})$ is planar. Then any cycle-minor $\mathcal{C}'$ of $\mathcal{C}$ on a weighted vertex set V satisfies the following:

(i) $\sum_{v \in V}$ weight$(v) = |V|$,

(ii) if weight$(v) = 0$ then vertex v is isolated,

(iii) if every vertex in $\mathcal{G}(\mathcal{C}^*)$ is either isolated or is a loop vertex, then for every self-loop (v, v) in $\mathcal{G}(\mathcal{C}^*)$ with multiplicity $k \geq 1$, it holds that weight$(v) \geq k$.

Proof: We first observe that any operation of cycle deletion does not change the total weight of the vertex set, which yields the first property. Further, any cycle contraction at a vertex decreases the weight of a vertex v only if v becomes isolated; in that case weight(v) becomes 0, what yields the second property.

Now we prove the third property. Observe that without loss of generality, we can obtain the cycle-minor $\mathcal{C}'$ of $\mathcal{C}$ on a weighted vertex set V by first deleting some cycles and then performing only contractions. Let us fix a self-loop (v, v) and let us consider the sequence of contractions that lead to the creation of k copies of (v, v) (since these contractions are performed after the cycle deletions, we can consider contractions for each self-loop separately). Each self-loop (v, v) corresponds to some cycle in $\mathcal{C}$ and let $\mathcal{C}_{(v,v)}$ be the set of the k cycles in $\mathcal{C}$ corresponding to the k self-loops (v, v). Let $V_{(v,v)}$ be the set of vertices induced by $\mathcal{C}_{(v,v)}$ in $\mathcal{G}(\mathcal{C})$. We observe that the final weight of vertex v in the cycle-minor $\mathcal{C}'$ is equal to the weight of $V_{(v,v)}$ in $\mathcal{C}$, and hence it is equal to $|V_{(v,v)}|$. Next, we notice that $\mathcal{G}(\mathcal{C}_{(v,v)})$ has at least $3|\mathcal{C}_{(v,v)}| = 3k$ edges (because each cycle is of length at least 3) and all cycles in $\mathcal{C}_{(v,v)}$ are edge-disjoint. Therefore, planarity of $\mathcal{G}(\mathcal{C}_{(v,v)})$ implies that the number of edges in $\mathcal{G}(\mathcal{C}_{(v,v)})$ is at most 3 times the number of vertices in $\mathcal{G}(\mathcal{C}_{(v,v)})$, and hence $|V_{(v,v)}| \geq |\mathcal{C}_{(v,v)}| = k$. ■

5.3. Random walk invariant

The central notion in our analysis is that of being *random walk invariant*. For that, we begin with the description of an extension of our algorithm Random Bipartitness Exploration to multigraphs on weighted vertex sets and with labeled edges: Algorithm Random Bipartitness Exploration in multigraphs (RBEM).

> **Algorithm RBEM** (ε, l)**:**
> - Repeat $f(\varepsilon)$ times:
>   - Pick a random vertex v with probability proportional to its weight
>   - Starting from v, run a random walk of length l
>   - If the random walk found an odd cycle then **reject**
> - If none of the random walks found an odd cycle then **accept**

Similarly to the previous section, we will focus on the random walk part of the algorithm. In the following, whenever we refer to a random walk, we will assume that the starting vertex is chosen randomly with probability proportional to its weight.

Definition 9 (Random walk invariant): Let $r > 0$ be integral and let $\xi > 0$ be an arbitrary constant. Let $\mathcal{C}$ be a set of edge-labeled cycles on a weighted vertex set V and let $\mathcal{C}'$ be a cycle-minor of $\mathcal{C}$. $\mathcal{C}'$ is called (r, l, ξ)-random walk invariant with respect to $\mathcal{C}$ if

- for every l-step walk $\mathcal{E}' = \langle x_0, x_1, \dots x_l \rangle$ in $\mathcal{G}(\mathcal{C}')$ along edges with parities $\langle a_1, \dots, a_{l-1} \rangle$, if the probability that RBEM(ε, l) invoked on $\mathcal{G}(\mathcal{C}')$ chooses $\mathcal{E}'$ as its random walk is p, then the probability that RBEM(ε, rl) invoked on $\mathcal{G}(\mathcal{C})$ chooses a walk $\mathcal{E} = \langle y_0, y_1, \dots y_{rl} \rangle$ in $\mathcal{G}(\mathcal{C})$ that contains $\mathcal{E}'$ as a subwalk (with similar parities) is at least $\xi \cdot p$. Formally, there exists a set $I = \{i_0, i_1, \dots, i_l\} \subseteq \{0, 1, \dots, rl\}$ such that (1) $x_j = y_{i_j}$ for every $j \in I$, (2) the parity of edge (x_j, x_{j+1}) in $\mathcal{G}(\mathcal{C}')$ is the same as the parity of the path $\langle y_{i_j}, y_{i_j+1}, \dots, y_{i_{j+1}} \rangle$ in $\mathcal{G}(\mathcal{C})$, and (3) for every $j \in \{0, 1, \dots, i_l\} \setminus I$ the weight of y_j is 0.

Observe that the notion of random walk invariance is related to our claim in Lemma 5. Indeed, in Lemma 5 we showed that in order to analyze the random walk approach in the original graph G that is ε-far from bipartite, it is enough to analyze the random walk approach in the subgraph of G induced by a linear number of short odd-length cycles. Definition 9 extends this claim and states that in order to analyze the random walk approach in $\mathcal{G}(\mathcal{C})$ it is enough to analyze it in a cycle-minor of $\mathcal{C}$ that is (r, l, ξ)-random walk invariant with respect to $\mathcal{C}$ (r and ξ may depend on ε).

Furthermore, let us observe the following simple fact.

Observation 10: Let $\mathcal{C}$ be a set of edge-labeled cycles on a weighted vertex set V. If $\mathcal{C}'$ is a cycle-minor of $\mathcal{C}$ that is $(r, r'l, \xi)$-random walk invariant with respect to $\mathcal{C}$, and if $\mathcal{C}''$ is a cycle-minor of $\mathcal{C}'$ that is (r', l, ξ')-random walk invariant with respect to $\mathcal{C}'$, then $\mathcal{C}''$ is $(rr', l, \xi\xi')$-random walk invariant with respect to $\mathcal{C}$.

5.4. Outline of the analysis

We now give a more detailed and technical outline of the proof. We initially assign weight 1 to every vertex in V, and we assign label *odd* to every edge in $\mathcal{G}(\mathcal{C})$. Now, we want to perform the following reduction: In order to show that Random Bipartitness Exploration finds (with constant probability) an odd-length cycle in a set $\mathcal{C}$ of $\alpha(\varepsilon)n$ edge-disjoint odd-length cycles in G, each cycle in $\mathcal{C}$ having length at most $\ell(\varepsilon)$, it is enough to show that for some $r = r(\varepsilon)$, $l = l(\varepsilon)$ and $\xi = \xi(\varepsilon)$ there is a cycle-minor $\mathcal{C}^*$ of $\mathcal{C}$ that is (r, l, ξ)-random walk invariant with respect to $\mathcal{C}$, such that $\mathcal{C}^*$ has a very simple form: it consists of sufficiently many odd self-loops only. To conclude the analysis, we observe that a 1-step random walk rejects if and only if it finds one of the self-loops. Hence, a 1-step random walk rejects with a constant probability if and only if the total weight of the non-isolated vertices (loop vertices) in V is a constant fraction of the total weight of the vertices. Now, finally, by the Vertex-weight Lemma 8, to complete the proof it is enough to ensure that the number of cycles in $\mathcal{C}^*$ is $\Omega(|V|)$. (Indeed, the Vertex-weight Lemma implies that the total weight of the non-isolated vertices is at least of the order of the number of cycles in $\mathcal{C}^*$.)

In view of the outline above, the key part of our analysis is to find set $\mathcal{C}^*$ that satisfies the conditions described above. Let us give the intuition how we find $\mathcal{C}^*$: we construct a sequence $\mathcal{C}_N, \mathcal{C}_{N-1}, \dots, \mathcal{C}_1$ of sets of labeled cycles on weighted vertex sets V such that for values $r = r(\varepsilon), \xi = \xi(\varepsilon)$ that depend only on ε, the following holds:

- $\mathcal{C}_N = \mathcal{C}$, $N = \ell(\varepsilon)$, all edges in $\mathcal{G}(\mathcal{C})$ are odd, and the weighted vertex set V for $\mathcal{C}_N$ has the weight of every vertex equal to 1,
- each set $\mathcal{C}_k$ consists of cycles of length at most k,
- each set $\mathcal{C}_k$ is a cycle-minor of $\mathcal{C}_{k+1}$, and
- each $\mathcal{C}_k$ is (r, r^{k-1}, ξ)-random walk invariant with respect to $\mathcal{C}_{k+1}$.

The key part of our analysis now is to transform $\mathcal{C}_k$ into its cycle-minor $\mathcal{C}_{k-1}$ that satisfies the properties described above. We do this by using the process that we call cycle-shortening.

5.5. Shortening cycles

We consider an edge-disjoint set of odd-length cycles $\mathcal{C}_N = \mathcal{C}$ on a vertex set V such that the length of each cycle $\mathfrak{c} \in \mathcal{C}$ is at most $\ell(\varepsilon)$. We assign weight 1 to every vertex in V, and we assign label *odd* to every edge in $\mathcal{C}$.

Let us first describe a generic procedure of *cycle-shortening*:

Cycle-shortening(set $\mathcal{C}_k$ of cycles of length $\leq k$)

- Choose an appropriate subset $\mathcal{C}^*$ of $\mathcal{C}_k$ such that every cycle in $\mathcal{C}^*$ has a vertex that is either a loop vertex or is contractible.
- Perform cycle contraction at an appropriate independent set of contractible vertices to ensure that each cycle in $\mathcal{C}^*$ other than a self-loop reduces the number of edges.
- Return $\mathcal{C}_{k-1}$ to be the obtained set of cycles.

(The formulation of an "appropriate" subset $\mathcal{C}^* \subseteq \mathcal{C}_k$ and an "appropriate" independent set of contractible vertices is used here in a generic sense and the exact choices of these sets are explained in the analysis.)

The following is a direct implication of our construction.

Observation 11: If we start with $\mathcal{C}$ being a set of odd-length cycles with the length of each cycle $\mathfrak{c} \in \mathcal{C}$ being at most $\ell(\varepsilon)$, then after applying cycle-shortening $\ell(\varepsilon) - 1$ times, we obtain a set of edge-labeled cycles $\mathcal{C}_1$ that is a cycle-minor of $\mathcal{C}$ and such that each cycle in $\mathcal{C}_1$ is an *odd* self-loop.

For our analysis, we want to have a construction of cycle-shortening that additionally has two central properties: the number of cycles in $\mathcal{C}_1$ should be not much smaller than the number of cycles in $\mathcal{C}$ (smaller only by a constant factor depending on ε), and there is an $r^* = r^*(\varepsilon)$ such that $\mathcal{C}_1$ is $(r^*, 1, \xi)$-random walk invariant with respect to $\mathcal{C}$, for a value $\xi = \xi(\varepsilon)$ depending only on ε.

The following is our central technical lemma.

Lemma 12: Let $\mathcal{C}_k$ be a set of edge-labeled cycles on a weighted vertex set V, where each cycle in $\mathcal{C}_k$ has length at most $k \leq \ell(\varepsilon)$ and such that the weighted graph representation of $\mathcal{G}(\mathcal{C})$ is planar. One can design a procedure **Cycle-shortening**($\mathcal{C}_k$) with output $\mathcal{C}_{k-1}$ such that there are r, ξ and ζ depending only on ε, for which the following holds:

- the number of cycles $\mathcal{C}_{k-1}$ is at least ζ times the number of cycles $\mathcal{C}_k$, and
- $\mathcal{C}_{k-1}$ is (r, r^{k-1}, ξ)-random walk invariant with respect to $\mathcal{C}_k$.

We observe that by our discussion above, Observation 10 and Lemma 12 directly imply Theorem 4, and hence proves the main result of the paper. We continue with the following lemma whose proofs is deferred to the full version.

Lemma 13: Let $\mathcal{C}$ be a set of edge-labeled cycles on a vertex set V, with each cycle in $\mathcal{C}$ having length at most k and such that the weighted graph representation of $\mathcal{G}(\mathcal{C})$ is planar. Then there is a subset $\mathcal{C}^* \subseteq \mathcal{C}$ and an independent set $Q \subseteq V$ in $\mathcal{G}(\mathcal{C}^*)$ such that *(i)* every vertex from Q is a contractible vertex with respect to $\mathcal{C}^*$, *(ii)* every cycle $\mathfrak{c} \in \mathcal{C}^*$ has a contractible vertex from Q in $\mathcal{G}(\mathcal{C}^*)$, *(iii)* after contracting at any vertex at Q to obtain new edge(s) (x, y), the pair $\langle x, y \rangle$ is τ-parity balanced for the obtained set of cycles, *(iv)* every vertex in $\mathcal{G}(\mathcal{C}^*)$ is either isolated or has degree at least ϑ times its degree in $\mathcal{G}(\mathcal{C})$, and *(v)* $|\mathcal{C}^*| \geq \sigma|\mathcal{C}|$, where τ, ϑ, σ are some values depending only on k.

With such a lemma, we can proceed with our algorithm:

Cycle-shortening(set $\mathcal{C}_k$ of edge-labeled cycles)
- Let $\mathcal{C}^*$ be the set of cycles and Q contractible vertices resulting from applying Lemma 13 to $\mathcal{C}_k$
- Perform cycle contraction at contractible vertices in Q
- Return $\mathcal{C}_{k-1}$ to be the obtained set of cycles

We will now move to our final technical lemma that analyzes properties of the algorithm **Cycle-shortening**.

Lemma 14: Let $\mathcal{C}$ be a set of edge-labeled cycles on a weighted vertex set V, with each cycle in $\mathcal{C}_k$ having length at most k and such that the weighted graph representation of $\mathcal{G}(\mathcal{C})$ is planar. Then the set of cycles $\mathcal{C}_{k-1}$ obtained after applying **Cycle-shortening**($\mathcal{C}_k$) satisfies the following conditions:

- $|\mathcal{C}_{k-1}| \geq \sigma|\mathcal{C}_k|$, where σ is a constant dependent only k, and
- there is $\xi = \xi(k)$ such that $\mathcal{C}_{k-1}$ is (r, r^{k-1}, ξ)-random walk invariant with respect to $\mathcal{C}_k$, where $r = 3$.

Proof: Let us note that the first part of the lemma follows directly from Lemma 13 and hence our main focus is on proving the second part of the lemma. We will be using the notation from Lemma 13.

Our proof will give $\xi = \xi(k,l) = \vartheta^{2l+1}\left(\frac{\tau}{1+\tau}\right)^l$.

Let us fix an arbitrary walk $\langle x_0, x_1, \ldots, x_l\rangle$ in $\mathcal{G}(\mathcal{C}_{k-1})$. Our goal is to compare the probability of having RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ to choose the walk $\langle x_0, x_1, \ldots, x_l\rangle$ and the probability that RBEM on $\mathcal{G}(\mathcal{C}_k)$ will choose a walk with a prefix of the form $\langle ?, x_0, ?, x_1, ?, x_2, \ldots, x_{l-1}, ?, x_l\rangle$, where we use the mark ? to denote either a vertex from Q or nothing, and we require that for any $0 \leq i < r$, both (x_i, x_{i+1}) and $\langle x_i, ?, x_{i+1}\rangle$ will have the same parity. (And so, for example, $\langle ?, x_0, ?, x_1\rangle$ denotes one of the following: path $\langle x_0, x_1\rangle$, or path $\langle v, x_0, x_1\rangle$ with an arbitrary vertex $v \in Q$, or path $\langle x_0, u, x_1\rangle$ with an arbitrary vertex $u \in Q$, or path $\langle v, x_0, u, x_1\rangle$ with an arbitrary pair of vertices $v, u \in Q$.) Such a prefix has length at most $2l + 1 \leq rl$ and therefore it may occur as the prefix of an rl-step random walk.

Let us introduce some notation. We will use $\mathcal{C}^*$ to denote the set obtained by applying Lemma 13 on $\mathcal{C}_k$. For any vertex $v \in V$, let Ψ_v be the set of contractible vertices in Q that are adjacent to v in $\mathcal{G}(\mathcal{C}^*)$, let A_v be the subset of Ψ_v that consists of vertices that have two distinct neighbors in $\mathcal{G}(\mathcal{C}^*)$, and let B_v be the subset of Ψ_v that consists of vertices that have a (one) unique neighbor in $\mathcal{G}(\mathcal{C}^*)$. Clearly, A_v and B_v form a partition of Ψ_v. We consider the set $\mathcal{C}_k$ of edge-labeled cycles on a weighted vertex set V and the set $\mathcal{C}_{k-1}$ of edge-labeled cycles on a weighted vertex set V; let weight(v) to be the weight of vertex $v \in V$ for $\mathcal{C}_k$ and let weight$^*(v)$ to be the weight of vertex $v \in V$ for $\mathcal{C}_{k-1}$. Notice that for every non-isolated vertex v in $\mathcal{C}_{k-1}$ we have:

$$\begin{aligned}\text{weight}^*(v) &= \\ &= \text{weight}(v) + \frac{1}{2}\sum_{u \in A_v} \text{weight}(u) + \sum_{u \in B_v} \text{weight}(u) \ .\end{aligned}$$

Furthermore, for any $v \in V$, let d_v be the degree of v in $\mathcal{G}(\mathcal{C}_k)$, d^*_v be the degree of v in $\mathcal{G}(\mathcal{C}^*)$, and D_v be the degree of v in $\mathcal{G}(\mathcal{C}_{k-1})$. Observe that by Lemma 13, for every $v \in V$, either $d^*_v = 0$ or $d^*_v \geq \vartheta d_v$ for a constant ϑ that depends on k. Furthermore, if $v \in Q$, then $D_v = 0$, and if $D_v > 0$ then $D_v = d^*_v$.

Starting the walk: A necessary condition for RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ to visit $\langle x_0, x_1, \ldots, x_l\rangle$ as a random walk starting at x_0 is that x_0 is chosen (at random) as the starting vertex of one of the random walks. Let us fix a single random walk.

We will couple this event with one of the corresponding events for RBEM on $\mathcal{G}(\mathcal{C}_k)$:

(i) x_0 is chosen as the starting vertex of the fixed random walk by RBEM on $\mathcal{G}(\mathcal{C}_k)$,

(ii) a vertex $u \in \Psi_{x_0}$ is chosen as the starting vertex of the fixed random walk by RBEM on $\mathcal{G}(\mathcal{C}_k)$ and then the random walk starts at u and moves to x_0 in a single step.

Observe that the first event will happen with the probability proportional to weight(v) and the second type of events will happen with the probability $\frac{1}{|V|}\sum_{u \in \Psi_{x_0}} \text{weight}(u) \cdot \frac{\text{multiplicity of edge } (u,v) \text{ in } \mathcal{G}(\mathcal{C}_k)}{d_u}$. Next, we observe that for any $u \in A_{x_0}$ the multiplicity of edge (u, v) is equal to $\frac{1}{2}d^*_u$, and for any $u \in B_{x_0}$ the multiplicity of edge (u, v) is equal to d^*_u. Therefore, since Lemma 13 ensures that $d^*_u \geq \vartheta d_u$, we can show that the probability that the event will happen for RBEM on $\mathcal{G}(\mathcal{C})$ is lower bounded by $\frac{\vartheta \cdot \text{weight}^*(v)}{|V|}$.

Therefore, to summarize, the probability that RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ will start at x_0 is at most $\frac{1}{\vartheta}$ greater than the probability that RBEM on $\mathcal{G}(\mathcal{C}_k)$ will reach x_0 in zero or one step.

Continuing the walk: Next, we assume that the walk reached vertex x_i in both RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ and RBEM on $\mathcal{G}(\mathcal{C}_k)$, and we compare the probability that RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ will take edge (x_i, x_{i+1}) with label χ_i (χ_i is either *odd* or *even*) vs. the probability that RBEM on $\mathcal{G}(\mathcal{C}_k)$ will take $\langle x_i?, x_{i+1}\rangle$ with label χ_i.

Let m_i be the number of edges (multiplicity of) (x_i, x_{i+1}) in $\mathcal{G}(\mathcal{C}^*)$ with label χ_i. For any $v \in Q$, let $k_i^{\langle v\rangle}$ be the number of paths $\langle x_i, v, x_{i+1}\rangle$ in $\mathcal{G}(\mathcal{C}^*)$ with label χ_i. Let $k_i = \sum_{v \in Q} k_i^{\langle v\rangle}$. Note that $m_i + k_i$ is exactly equal to the multiplicity of edge (x_i, x_{i+1}) in $\mathcal{G}(\mathcal{C}_{k-1})$ with label χ_i.

Let us first consider RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$. We observe that if it starts at x_i, then it chooses edge (x_i, x_{i+1}) with label χ_i with probability $\frac{m_i+k_i}{D_{x_i}}$.

Next, let us consider RBEM on $\mathcal{G}(\mathcal{C}_k)$. If it starts at x_i, then one way to proceed is if it will choose edge (x_i, x_{i+1}) with label χ_i; this will happen with probability at least $\frac{m_i}{d_{x_i}}$ (because every edge from $\mathcal{C}^*$ is also present in $\mathcal{C}_k$ with the same label, and so there are at least m_i edges (x_i, x_{i+1}) in $\mathcal{G}(\mathcal{C}_k)$ with label χ_i). If it starts at x_i, then we can also take a path $\langle x_i, v, x_{i+1}\rangle$ with label χ_i and with $v \in Q$; this will happen with the probability at least $\frac{k_i^{\langle v\rangle}}{d_{x_i}} \cdot \frac{k_i^{\langle v\rangle}}{d_v}$ (and assuming that $d_v > 0$). Therefore, the probability that when starting at x_i, RBEM on $\mathcal{G}(\mathcal{C}_k)$ will take $\langle x_i ?, x_{i+1}\rangle$ with label χ_i is at least $\frac{m_i}{d_{x_i}} + \sum_{v \in Q, d_v^* \neq 0} \frac{k_i^{\langle v\rangle}}{d_{x_i}} \cdot \frac{k_i^{\langle v\rangle}}{d_v}$.

Now, by Lemma 13, we know that $D_{x_i} = d_{x_i}^* \geq \vartheta d_{x_i}$ and that $d_v^* \geq \vartheta d_v$ for every $v \in Q$. Furthermore, since the pair $\langle x_i, x_{i+1}\rangle$ is τ-parity balanced, for every $v \in Q$, we either have $d_v^* = 0$ or $\frac{k_i^{\langle v\rangle}}{d_v^*} \geq \frac{\tau}{1+\tau}$. Therefore we obtain that the probability that when starting at x_i, RBEM on $\mathcal{G}(\mathcal{C}_k)$ will take $\langle x_i ?, x_{i+1}\rangle$ with label χ_i can be shown to be lower bounded as follows:

$$\frac{m_i}{d_{x_i}} + \sum_{v \in Q} \frac{k_i^{\langle v\rangle}}{d_{x_i}} \cdot \frac{k_i^{\langle v\rangle}}{d_v} \quad \geq \quad \frac{\vartheta^2 \tau}{1+\tau} \cdot \frac{(m_i + k_i)}{D_{x_i}} \quad .$$

Therefore, conditioned on the walk starting at vertex x_i, the probability that RBEM on $\mathcal{G}(\mathcal{C}_k)$ will take $\langle x_i ?, x_{i+1}\rangle$ with label χ_i is at least $\frac{\vartheta^2 \tau (m_i + k_i)}{(1+\tau) D_{x_i}}$, which is at least $\frac{\vartheta^2 \tau}{1+\tau}$ times the probability that RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ will take edge (x_i, x_{i+1}) with label χ_i.

We can summarize our discussion above. Let us fix an arbitrary walk $\langle x_0, x_1, \dots, x_l\rangle$ in $\mathcal{G}(\mathcal{C}_{k-1})$ and let χ be the label of the walk. Then, by our arguments above, the probability that RBEM on $\mathcal{G}(\mathcal{C})$ will choose a walk with prefix $\langle ?, x_0, ?, x_1, ?, x_2, \dots, x_{r-1}, ?, x_r\rangle$ labeled χ is at least $\vartheta \cdot \left(\frac{\vartheta^2 \tau}{1+\tau}\right)^l = \vartheta^{2l+1} \cdot (\frac{\tau}{1+\tau})^l$ times the probability that RBEM on $\mathcal{G}(\mathcal{C}_{k-1})$ will choose the walk $\langle x_0, x_1, \dots, x_l\rangle$ with label χ. ■

6. Further Research

In this paper we proved that bipartiteness is testable in constant time for arbitrary planar graphs. Our result was proven via a new type of analysis of random walks in planar graphs. Our analysis easily carries over to classes of graphs defined by arbitrary fixed forbidden minors.

This is merely the first step that poses the following main question:

What graph properties can be tested in constant time in minor-free graphs?

References

[1] N. Alon, E. Fischer, I. Newman, and A. Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *In Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)* pp. 251–260, 2006.

[2] N. Alon and M. Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Mathematics*, 15(2):211-227, 2002.

[3] I. Benjamini, O. Schramm, and A. Shapira. Every minor-closed property of sparse graphs is testable. *In Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 393–402, 2008.

[4] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. *In Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 93–102, 2002.

[5] A. Czumaj, A. Shapira, and C. Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6): 2499–2510, April 2009.

[6] A. Czumaj and C. Sohler. Testing expansion in bounded-degree graphs. *In Proceedings of the 48th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 570–578, 2007.

[7] O. Goldreich, editor. *Property Testing: Current Research and Surveys*. Lecture Notes in Computer Science 6390, Springer Verlag, Berlin, Heidelberg, December 2010.

[8] O. Goldreich, S. Goldwasser, D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4): 653–750, July 1998.

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

[10] O. Goldreich and D. Ron. A sublinear bipartiteness tester for bounded degree graphs. *Combinatorica*, 19(3):335–373, 1999.

[11] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, Report No. 7, 2000.

[12] A. Hassidim, J. A. Kelner, H. N. Nguyen, and K. Onak. Local graph partitions for approximation and testing. *In Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 22–31, 2009.

[13] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6): 1441–1483, September 2004.

[14] P. Klein, S. Plotkin, S. Rao. Excluded minors, network decomposition, and multicommodity flow. *In Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 682–690, 1993.

[15] S. Marko and D. Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Transactions on Algorithms*, 5(2), Article No. 22, March 2009.

[16] I. Newman and C. Sohler. Every property of hyperfinite graphs is testable. *In Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 675–684, 2011.

[17] H. N. Nguyen and K. Onak. Constant-time approximation algorithms via local improvements. *In Proceedings of the 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 327–336, 2008.