Testable Bounded Degree Graph Properties Are Random Order Streamable

Morteza Monemizadeh*[1], S. Muthukrishnan[2], Pan Peng†[3], and Christian Sohler‡[4]

1 Department of Computer Science, Goethe-Universität Frankfurt, Frankfurt, Germany
monemi@ae.cs.uni-frankfurt.de

2 Rutgers University, Piscataway, NJ, USA
muthu@cs.rutgers.edu

3 University of Vienna, Faculty of Computer Science, Vienna, Austria
pan.peng@univie.ac.at

4 Department of Computer Science, TU Dortmund, Dortmund, Germany
christian.sohler@tu-dortmund.de

Abstract

We study which property testing and sublinear time algorithms can be transformed into graph streaming algorithms for random order streams. Our main result is that for bounded degree graphs, any property that is constant-query testable in the adjacency list model can be tested with *constant space* in a single-pass in random order streams. Our result is obtained by estimating the distribution of local neighborhoods of the vertices on a random order graph stream using constant space.

We then show that our approach can also be applied to constant time approximation algorithms for bounded degree graphs in the adjacency list model: As an example, we obtain a constant-space single-pass random order streaming algorithms for approximating the size of a maximum matching with additive error ϵn (n is the number of nodes).

Our result establishes for the first time that a large class of sublinear algorithms can be simulated in random order streams, while $\Omega(n)$ space is needed for many graph streaming problems for adversarial orders.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Graph streaming algorithms, graph property testing, constant-time approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.131

1 Introduction

Very large and complex networks abound. Some of the prominent examples are gene regulatory networks, health/disease networks, and online social networks like Facebook, Google+, Linkedin and Twitter. The interconnectivity of neurons in human brain, relations in database systems, and chip designs are some further examples. Some of these networks can be quite large and it may be hard to store them completely in the main memory and some

* Partially supported by DFG grants ME 2088/3-(1/2) and ME 2088/4-1.
† The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 340506.
‡ Supported by ERC Starting Grant 307696.

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 131; pp. 131:1–131:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

may be too large to be stored at all. However, these networks contain valuable information that we want to reveal. For example, social networks can provide insights into the structure of our society, and the structure in gene regulatory networks might yield insights into diseases. Thus, we need algorithms that can analyze the structure of these networks quickly.

One way to approach this problem is to design graph streaming algorithms [16, 1]. A graph streaming algorithm gets access to a stream of edges in some order and exactly or approximately solves problems on the graph defined by the stream. The challenge is that a graph streaming algorithm should use space sublinear in the size of the graph. We focus on algorithms that make only *one pass* over the graph stream. It has been shown that many natural graph problems require $\Omega(n)$ space in the *adversarial order* model where n is the number of nodes in the graph and the edges can arrive in arbitrary order (see eg.,[8, 9]), and thus most of previous work has focused on the *semi-streaming* model, in which the algorithms are allowed to use $O(n \cdot \operatorname{poly}\log n)$ space. However, in many interesting applications, the graphs are sparse and so they can be fully stored in the semi-streaming model making this model useless in this setting. This raises the question *whether there are at least some natural conditions under which one can solve graph problems with space $o(n)$, possibly even $\log^{O(1)} n$ or constant.*

One such condition that recently received increasing attention is that the edges arrive in *random order*, i.e. in the order of a uniformly random permutation of the edges (e.g., [5, 22, 19]). Uniformly random or near-uniformly random ordering is a natural assumption and can arise in many contexts. Indeed, previous work has shown that some problems that are hard for adversarial streams can be solved in the random order model. Konrad et al. [22] gave single-pass semi-streaming algorithms for maximum matching for bipartite and general graphs with approximation ratio strictly larger than $1/2$ in the random order semi-streaming model, while no such approximation algorithm is known in the adversary order model. Kapralov et al. [19] gave a polylogarithmic approximation algorithm in polylogarithmic space for estimating the size of maximum matching of an unweighted graph in one pass over a random order stream. Assadi et al. [2] recently showed that in the adversarial order and *dynamic* model where edges can be both inserted and deleted, any polylogarithmic approximation algorithm of maximum matching size requires $\tilde{\Omega}(n)$ space. On the other hand, Chakrabarti et al. [5] presented an $\Omega(n)$ space lower bound for any single pass algorithm for graph connectivity in the random order streaming model, which is very close to the optimal $\Omega(n \log n)$ space lower bound in the adversarial order model [30]. In general, it is unclear which graph problems can be solved in random order streams using much smaller space than what is required for adversarially ordered streams.

An independent area of research is *property testing*, where with certain *query* access to an object (eg., random vertices or neighbors of a vertex for graphs), there are algorithms that can determine if the object satisfies a certain property, or is far from having such a property [29, 11, 12]. The area of property testing has seen fundamental results, including testing various general graph properties. For example, it has been shown that many interesting properties (including connectivity, planarity, minor-freeness, hyperfiniteness) of bounded degree graphs can be tested with a constant number of queries [12, 3, 25]. Another very related area of research is called *constant-time* (or in general, *sublinear-time*) *approximation* algorithms, where we are given query access to an object (for example a graph) and the goal is to approximate the objective value of an optimal solution. For example, in bounded degree graphs, one can approximate the cost of the optimal solution with constant query complexity for some fundamental optimization problems (e.g., minimum spanning tree weight [6], maximal matching size [26]; see also Section 1.3).

A fundamental question is if such results from property testing and constant-time approximation algorithms will lead to better graph streaming algorithms. Huang and Peng [17] recently considered the problem of estimating the minimum spanning tree weight and property testing for general graphs in dynamic and adversarial order model. They showed that a number of properties (e.g., connectivity, cycle-freeness) of general n-vertex graphs can be tested with space complexity $O(n^{1-\varepsilon})$ and one can $(1+\varepsilon)$-approximate the weight of minimum spanning tree with similar space guarantee. Furthermore, there exist $\Omega(n^{1-O(\varepsilon)})$ space lower bounds for these problems that hold even in the insertion-only model [17].

1.1 Overview of Results

In this paper we provide a general framework that transforms bounded-degree graph property testing to very space-efficient random order streaming algorithms.

To formally state our main result, we first review some basic definitions of graph property testing. A *graph property* is a property that is invariant under graph isomorphism. Let $G = (V, E)$ be a graph with maximum degree upper bounded by a constant d, and we also call G a *d-bounded* graph. In the *adjacency list model* for (bounded-degree) graph property testing, we are given query access to the adjacency list of the input d-bounded graph $G = (V, E)$. That is, for any vertex $v \in V$ and index $i \leq d$, one can query the ith neighbor (if exists) of vertex v in constant time. Given a property Π, we are interested in testing if a graph G satisfies Π or is *ε-far* from satisfying Π while making as few queries as possible, where G is said to be ε-far from satisfying Π if one has to insert/delete more than εdn edges to make it satisfy Π. We call a property *constant-query testable* if there exists a testing algorithm (also called *tester*) for this property such that the number of performed queries depends only on parameters ε, d and is independent of the size of the input graph.

Given a graph property Π, we are interested in *approximately* testing it in a single-pass stream with a goal similar to the above. That is, the algorithm uses little space and with high constant probability, it accepts the input graph G if it satisfies P and rejects G if it is ε-far from satisfying P (see Section 4 for formal definitions). Our main result is as follows.

▶ **Theorem 1.** *Any d-bounded graph property that is constant-query testable in the adjacency list model can be tested in the uniformly random order streaming model with constant space.*

To the best of our knowledge, this is the first non-trivial graph streaming algorithm with constant space complexity (measured in the number of *words*, where a word is a space unit large enough to encode an ID of any vertex in the graph.) By the constructions in [17], there exist graph properties (e.g., connectivity and cycle-freeness) of d-bounded graphs such that any single-pass streaming algorithm in the insertion-only and *adversary order* model must use $\Omega(n^{1-O(\varepsilon)})$ space. In contrast to this lower bound, our main result implies that d-bounded connectivity and cycle-freeness can be tested in constant space in the random order stream model, since they are constant-query testable in the adjacency list model [12].

Our approach also works for simulating *constant-time approximation* algorithms as graph streaming algorithms with constant space. For a minimization (resp., maximization) optimization problem P and an instance I, we let $\mathrm{OPT}(I)$ denote the value of some optimal solution of I. We call a value x an (α, β)-approximation for the problem P, if for any instance I, it holds that $\mathrm{OPT}(I) \leq x \leq \alpha \cdot \mathrm{OPT}(I) + \beta$ (resp., $\frac{\mathrm{OPT}(I)}{\alpha} - \beta \leq x \leq \mathrm{OPT}(I)$). For example, it is known that there exists a constant-query algorithm for $(1, \varepsilon n)$-approximating the maximal matching size of any n-vertex d-bounded graph [26]. That is, the number of queries made by the algorithm is independent of n and only depends on ε, d. As an application, we show:

▶ **Theorem 2.** *Let $0 < \varepsilon < 1$ and d be constants. Then there exists an algorithm that uses constant space in the random order model, and with probability $2/3$, $(1, \varepsilon n)$-approximates the size of some maximal matching in d-bounded graphs.*

We also remark that in a similar way, many other sublinear time algorithms for bounded degree graphs can be simulated in random order streams. Finally, our results can actually be extended to a model which requires weaker assumptions on the randomness of the order of edges in the stream, but we describe our results for the uniformly random order model, and leave the remaining details for later.

1.2 Technical Overview

The local neighborhood of depth k of a vertex v is the subgraph rooted at v and induced by all vertices of distance at most k from v. We call such a rooted subgraph a *k-disc*. Suppose that we are given a sufficiently large graph G whose maximum degree d is constant. This means that for any constant k, a k-disc centered at an arbitrary vertex v in G has constant size. Now assume that there exists an algorithm $\mathcal{A}$ that, independent of the labeling of the vertices of G, accesses G by querying random vertices and exploring their k-discs. We observe that any constant-query property tester (see for example [13, 7]) falls within the framework of such an algorithm. If instead of the graph G we are given the distribution of k-discs of the vertices of G, we can use this distribution to simulate the algorithm $\mathcal{A}$ and output with high probability the same result as executing the algorithm $\mathcal{A}$ on G itself. Thus, the problem of developing constant-query property testers in random order streams can be reduced to the problem of designing streaming algorithms that approximate the distribution of k-discs in G.

The main technical contribution of this paper is an algorithm that given a random order stream S of edges of an underlying d-bounded degree graph G, approximates the distribution of k-discs of G up to an additive error of δ. We would like to mention that if the edges arrive in adversarial order, any algorithm that approximates the distribution of k-discs of G requires almost linear space [32, 17], hence the assumption of random order streams (or something similar) is necessary to obtain our result.

Now in order to approximate the distribution of k-discs of the graph G we do the following. We proceed by sampling vertices uniformly at random and then perform a BFS for each sampled vertex using the arrival of edges along the stream S. Note that the new edges of the stream S that do not connect to the currently explored vertices are discarded. Let us call the k-disc that is observed by doing such a BFS from some vertex v to be Δ_1. Due to possibility of missing edges during the BFS, this subgraph may be different from the true k-disc Δ_2 rooted at v.

Fortunately, since the edges arrive in a uniformly random order, we can infer the conditional probability $\Pr[\Delta_1|\Delta_2]$. That is, given the true rooted subgraph Δ_2, we can compute the conditional probability of seeing a rooted subgraph Δ_1 in a random order stream when the true k-disc is Δ_2.

We define the partial order on the set of k-discs given by $\Delta_1 \preccurlyeq \Delta_2$ whenever Δ_1 is a root-preserving isomorphic subgraph of Δ_2. For every two k-discs Δ_1 and Δ_2 with $\Delta_1 \preccurlyeq \Delta_2$ we compute the conditional probability $\Pr[\Delta_1|\Delta_2]$. Using the set of all conditional probabilities $\Pr[\Delta_1|\Delta_2]$ we can estimate or approximate the distribution of k-discs of the graph G whose edges are revealed according to the stream S. In order to simplify the analysis of our algorithm, we require a natural independence condition for non-intersecting k-discs. Finally, we use the approximated distribution of k-discs to simulate the algorithm $\mathcal{A}$ by the machinary that we explained above.

We remark that the idea of using a partial order to compute a distribution of k-discs in bounded degree graphs has first been used in [7]. However, the setting in [7] was quite different as it dealt with directed graphs where an edge can only be seen from one side (and the sample sizes required in that paper were only slightly sublinear in n).

1.3 Other Related Work

Feigenbaum et al. [10] initiated the study of property testing in streaming model, and they gave efficient testers for some properties of a sequence of data items (rather than graphs as we consider here). Bury and Schwiegelshohn [4] gave a lower bound of $n^{1-O(\varepsilon)}$ on the space complexity of any algorithm that $(1-\varepsilon)$-approximates the size of maximum matching in adversarial streams. Kapralov et al. [20] showed that in random streams, $\tilde{\Omega}(\sqrt{n})$ space is necessary to distinguish if a graph is bipartite or 1/2-far from being bipartite. Previous work has extensively studied streaming graph algorithms in both the insertion-only and dynamic models, see the recent survey [24].

In the framework of d-bounded graph property testing, it is now known that many interesting properties are constant-query testable in the adjacency list model, including k-edge connectivity, cycle-freeness, subgraph-freeness [12], k-vertex connectivity [33], minor-freeness [15, 3], matroids related properties [18, 31], hyperfinite properties [25], subdivision-freeness [21]. Constant-time approximation algorithms in d-bounded graphs are known to exist for a number of fundamental optimization problems, including $(1+\varepsilon)$-approximating the weight of minimum spanning tree [6], $(1, \varepsilon n)$-approximating the size of maximal/maximum matching [26, 34], $(2, \varepsilon n)$-approximating the minimum vertex cover size [28, 23, 27], $(O(\log d), \varepsilon n)$-approximating the minimum dominating set size [28, 26]. For d-bounded minor-free graphs, there are constant-time $(1, \varepsilon n)$-approximation algorithms for the size of minimum vertex cover, minimum dominating set and maximum independent set [15].

2 Preliminaries

Let $G = (V, E)$ be an n-vertex graph with maximum degree upper bounded by some constant d, where we often identify V as $[n] := \{1, \cdots, n\}$. We also call such a graph d-bounded graph. In this paper, we will assume the algorithms have the knowledge of n, d. We assume that G is represented as a sequence of edges, which we denote as $\textsc{Stream}(G)$.

Graph k-discs. Let $k \geq 1$. The k-disc around a vertex v is the subgraph rooted at vertex v and induced by the vertices within distance at most k from v. Note that for an n-vertex graph, there are exactly n k-discs. Let $\mathcal{H}_{d,k} = \{\Delta_1, \cdots, \Delta_N\}$ be the set of all k-disc isomorphism types, where $N = N_{d,k}$ is the number of all such types (and is thus a constant). In the following, we will refer to a k-disc of some vertex v in the graph G as $\mathrm{disc}_{k,G}(v)$ and a k-disc type as Δ. Note that for every vertex v, there exists a unique k-disc type $\Delta \in \mathcal{H}_{d,k}$ such that $\mathrm{disc}_{k,G}(v)$ is isomorphic to Δ, denoted as $\mathrm{disc}_{k,G}(v) \cong \Delta$. (Throughout the paper, we call two rooted graphs H_1, H_2 isomorphic to each other if there is a root-preserving mapping from the vertex set of H_1 to the vertex set of H_2.)

We further assume that all the elements in $\mathcal{H}_{d,k}$ are ordered according to the natural partial order among k-disc types. More specifically, for any two k-disc types Δ_i, Δ_j, we let $\Delta_i \succcurlyeq \Delta_j$ (or equivalently, $\Delta_j \preccurlyeq \Delta_i$) denote that Δ_j is root-preserving isomorphic to some subgraph of Δ_i. Then we order all the k-disc types $\Delta_1, \cdots, \Delta_N$ such that if $\Delta_i \succcurlyeq \Delta_j$, then $i \leq j$. Let $\mathcal{G}(j)$ denote all the indices i, except j itself, such that $\Delta_i \succcurlyeq \Delta_j$.

Locally random order streams. Let Σ_E denote the set of all permutations (or orderings) over the edge set E. Note that each $\sigma \in \Sigma_E$ determines the order of edges arriving from the stream. Let $\mathcal{D} = \mathcal{D}(\Sigma_E)$ denote a probability distribution over Σ_E. In particular, we let $\mathcal{U} = \mathcal{U}(\Sigma_E)$ denote the uniform distribution over Σ_E. Given a stream σ of edges, we define the *observed k-disc of v from the stream*, denoted as $\mathrm{disc}_k(v, \sigma)$, to be the subgraph rooted at v and induced by all edges that are sequentially collected from the stream and the endpoints of which are within distance at most k to v. This is formally defined in the following algorithm Stream_k-disc.

Algorithm 1 The observed k-disc of v from the stream

```
procedure Stream_$k$-disc(Stream($G$),$k$,$v$)
    $U \leftarrow \{v\}$, $\ell_v = 0$, $F \leftarrow \emptyset$
    for $(u, w) \leftarrow$ next edge in the stream do
        if exactly one of $u, w$, say $u$, is contained in $U$ then
            if $\ell_u \leq k - 1$ then
                $U \leftarrow U \cup \{w\}, F \leftarrow F \cup \{(u, w)\}$
                for $x \in U$ do
                    $\ell_x \leftarrow$ the distance between $x$ and $v$ in the graph $G' = (U, F)$
                end for
            end if
        else if both $u, v$ are contained in $U$ then
            $F \leftarrow F \cup \{(u, w)\}$
            for $x \in U$ do
                $\ell_x \leftarrow$ the distance between $x$ and $v$ in the graph $G' = (U, F)$
            end for
        end if
    end for
    return $\mathrm{disc}_k(v, \sigma) \leftarrow$ the subgraph rooted at $v$ and induced by all edges in $F$
end procedure
```

Now we formally define a locally random distribution on the order of edges.

▶ **Definition 3.** Let $d, k > 0$. Let $G = (V, E)$ be a d-bounded graph. Let $\mathcal{D}$ be a distribution over all the orderings of edges in E. Let $\Lambda_k = \{\lambda(\Delta_i|\Delta_j) : 0 \leq \lambda(\Delta_i|\Delta_j) \leq 1, \Delta_j \not\succeq \Delta_i, 1 \leq i, j \leq N\}$ be a set of real numbers in $[0, 1]$. We call $\mathcal{D}$ a *locally random Λ_k-distribution* over G with respect to k-disc types, if for σ sampled from $\mathcal{D}$, the following conditions are satisfied:

1. (*Conditional probabilities*) For any vertex v with k-disc isomorphic to Δ_j, the probability that its observed k-disc $\mathrm{disc}_k(v, \sigma) \cong \Delta_i$ is $\lambda(\Delta_i|\Delta_j)$, for any i such that $\Delta_j \not\succeq \Delta_i$.
2. (*Independence of disjoint k-discs*) For any two disjoint k-discs $\mathrm{disc}_{k,G}(v)$ and $\mathrm{disc}_{k,G}(u)$, their observed k-discs $\mathrm{disc}_k(v, \sigma)$ and $\mathrm{disc}_k(u, \sigma)$ are independent.

Note that the set Λ_k cannot be an arbitrary set, as there might be no distribution satisfying the above condition. On the other hand, if there indeed exists a distribution satisfying the condition with numbers in Λ_k, then we call the set Λ_k *realizable*. In the following, we call a stream a *locally random order stream* if there exists a family of realizable sets $\Lambda = \{\Lambda_k\}_{k \geq 1}$, such that the edge order is sampled from some locally random Λ_k-distribution with respect to k-disc types, for any integer $k \geq 1$. We have the following lemma.

▶ **Lemma 4.** *Let $d \geq 1$. For any $k \geq 1$, there exists $n_0 = n_0(k, d)$, such that for $n \geq n_0$, any d-bounded n-vertex graph $G = (V, E)$, the uniform permutation $\mathcal{U}$ over E is a locally random Λ_k-distribution over G with respect to k-disc types, for some realizable $\Lambda_k := \{\lambda(\Delta_i|\Delta_j) : 0 \leq \lambda(\Delta_i|\Delta_j) \leq 1, \Delta_j \succcurlyeq \Delta_i, 1 \leq i, j \leq N\}$. Furthermore, if we let $\kappa := \max_{i,j:\Delta_j \succcurlyeq \Delta_i} \frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)}$, $\lambda_{\min} := \min_{i \leq N} \lambda(\Delta_i|\Delta_i)$, then $\kappa \leq 2^{2d^{k+1}}$, $\lambda_{\min} \geq \frac{1}{(2d^{k+1})!}$.*

Proof. Note that for any vertex v with $\mathrm{disc}_{k,G}(v) \cong \Delta_j$, the probability that the observed k-disc of v is isomorphic to Δ_i is exactly the fraction of orderings σ such that $\mathrm{disc}_k(v, \sigma) \cong \Delta_i$, where $\Delta_j \succcurlyeq \Delta_i$. We use such a fraction, which is a fixed real number, to define $\lambda(\Delta_i|\Delta_j)$. Observe that for an ordering σ sampled from $\mathcal{U}$, it directly satisfies the second condition Item 2 in Definition 3. Since there are at most $2d^{k+1}$ edges in any k-disc, the probability of observing a full k-disc is at least $\frac{1}{(2d^{k+1})!}$, that is, $\lambda_{\min} \geq \frac{1}{(2d^{k+1})!}$. Furthermore, since the k-disc type Δ_j might contain at most $\binom{|E(\Delta_j)|}{|E(\Delta_i)|} \leq 2^{2d^{k+1}}$ different subgraphs that are isomorphic to Δ_i, it holds that $\lambda(\Delta_i|\Delta_j) \leq \sum_{\substack{F:F \text{ subgraph of } \Delta_j \\ F \cong \Delta_i}} \lambda(\Delta_i|\Delta_i) \leq 2^{2d^{k+1}} \lambda(\Delta_i|\Delta_i)$ for any i, j such that $\Delta_j \succcurlyeq \Delta_i$. This completes the proof of the lemma. ◀

The above lemma shows that the uniformly random order stream is a special case of a locally random order stream. Another natural class of locally random order stream is ℓ-wise independent permutation of edges for any $\ell = \omega_n(1)$ (i.e., any function that tends to infinity as n goes to infinity) for n-vertex bounded degree graphs, but for our qualitative purposes here, it suffices to consider uniformly random order streams.

3 Approximating the k-Disc Type Distribution

In this section, we show how to approximate the distribution of k-disc types of any d-bounded graph in locally random order streams.

Recall that for any k, d, we let $N = N_{d,k}$ be the constant denoting the number of all possible k-disc isomorphism types. For any $i \leq N$, let V_i be the set of vertices from V with k-disc isomorphic to Δ_i in the input graph G, that is, $V_i := \{v | v \in V, \mathrm{disc}_{k,G}(v) \cong \Delta_i\}$. Note that $f_i = \frac{|V_i|}{n}$ is the fraction of vertices with k-disc isomorphic to Δ_i.

▶ **Lemma 5.** *Let $G = (V, E)$ be a d-bounded graph presented in a locally random order stream defined by a Λ_k-distribution $\mathcal{D}$ over G with respect to k-disc types, for some integer k. Let $\kappa := \max_{i,j:\Delta_j \succcurlyeq \Delta_i} \frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)}$, $\lambda_{\min} := \min_{i \leq N} \lambda(\Delta_i|\Delta_i)$. Then for any constant $\delta > 0$, there exists an algorithm that uses $O(\frac{\kappa^{2N} \cdot d^{3k+2} \cdot 3^{3N+1}}{\delta^2 \lambda_{\min}})$ space, and with probability $\frac{2}{3}$, for any $i \leq N$, approximates the fraction f_i of vertices with k-disc isomorphic to Δ_i in G with additive error δ.*

Proof. Our algorithm is as follows. We first sample a constant number of vertices, which are called centers. Then for each center v, we collect the observed k-disc of v from the stream. Then we postprocess all the collected edges and use the corresponding empirical distribution of k-disc types of all centers to estimate the distribution of k-disc types of the input graph. The formal description is given in Algorithm 2.

Note that since there are $s = \frac{8\kappa^{2N} \cdot d^{2k+1} \cdot 3^{3N+1}}{\delta^2 \lambda_{\min}}$ vertices in A and only edges that belong to the k-discs of these vertices will be collected by our algorithm, the space complexity of the algorithm is $O(sd^{k+1}) = O(\frac{\kappa^{2N} \cdot d^{3k+2} \cdot 3^{3N+1}}{\delta^2 \lambda_{\min}})$, which is constant.

Now we show the correctness of the algorithm.

Algorithm 2 Approximating the distribution of k-disc types

```
procedure $k$-DISC_DISTRIBUTION(STREAM($G$),$\Lambda_k$,$n, d, k, \delta$)
    sample a set $A$ of $s := \frac{8\kappa^{2N} \cdot d^{2k+1} \cdot 3^{3N+1}}{\delta^2 \lambda_{\min}}$ vertices uniformly at random
    for each $v \in A$ do
        $H_v \leftarrow$ STREAM_$k$-DISC(STREAM($G$),$v$,$k$)        ▷ to collect observed $k$-disc of $v$
    end for
end procedure

procedure POSTPROCESSING
    $H \leftarrow$ the graph spanned by $\cup_{v \in A} H_v$
    for $i = 1$ to $N$ do
        $Y_i \leftarrow |\{v : v \in A, \mathrm{disc}_{k,H}(v) \cong \Delta_i\}|/s$
        $X_i \leftarrow (Y_i - \sum_{j \in \mathcal{G}(i)} X_j \cdot \lambda(\Delta_i|\Delta_j)) \cdot \lambda^{-1}(\Delta_i|\Delta_i)$.
    end for
    return $X_1, \cdots, X_N$
end procedure
```

We let $A \sim \mathcal{U}_V$ denote that A is the set of s vertices sampled uniformly at random from V. For any $i \leq N$, let A_i be the set of vertices from A with k-disc isomorphic to Δ_i in the input graph G, that is, $A_i := \{v | v \in A, \mathrm{disc}_{k,G}(v) \cong \Delta_i\}$. Note that $\mathrm{E}_{A \sim \mathcal{U}_V}[|A_i|] = s \cdot \frac{|V_i|}{n}$.

Let $\beta_i = 3^{i-N-2}, \theta_i = (3\kappa)^{i-N-1}$. By Chernoff bound and our setting of s which satisfy that $s \geq \Omega(\frac{1}{(\delta\theta_i)^2 \beta_i})$, we have the following claim.

▶ **Claim 6.** *For any $i \leq N$, $\mathrm{Pr}_{A \sim \mathcal{U}_V}[|\frac{|A_i|}{s} - \frac{|V_i|}{n}| \leq \delta\theta_i] \geq 1 - \beta_i$.*

We assume for now that A is a fixed set with s vertices. We let $\sigma \sim \mathcal{D}$ denote that the edge ordering σ is sampled from $\mathcal{D}$. For any $v \in A$, let $Z_{v,i}$ be the indicator random variable of the event that the observed k-disc $\mathrm{disc}_k(v, \sigma)$ of v is isomorphic to Δ_i for $\sigma \sim \mathcal{D}$. Note that $\mathrm{Pr}_{\sigma \sim \mathcal{D}}[Z_{v,i} = 1] = \lambda(\Delta_i|\Delta_j)$ if $\mathrm{disc}_{k,G}(v) \cong \Delta_j$. Let $Y_i^{(\sigma)} := \frac{|\{v : v \in A, \mathrm{disc}_k(v,\sigma) \cong \Delta_i\}|}{s}$ denote the fraction of vertices in A with observed k-disc isomorphic to Δ_i. By definition, it holds that $Y_i^{(\sigma)} = \frac{1}{s} \sum_{\substack{v \in A_j \\ j \in \mathcal{G}(i) \cup \{i\}}} Z_{v,i}$, and furthermore, $\mathrm{E}_{\sigma \sim \mathcal{D}}[Y_i^{(\sigma)}] = \frac{1}{s} \sum_{j \in \mathcal{G}(i) \cup \{i\}} |A_j| \cdot \lambda(\Delta_i|\Delta_j)$. Let $X_i^{(\sigma)} = (Y_i^{(\sigma)} - \sum_{j \in \mathcal{G}(i)} X_j^{(\sigma)} \cdot \lambda(\Delta_i|\Delta_j)) \cdot \lambda^{-1}(\Delta_i|\Delta_i)$.

We have the following claim.

▶ **Claim 7.** *For any $i \leq N$, it holds that $E_{\sigma \sim \mathcal{D}}[X_i^{(\sigma)}] = \frac{|A_i|}{s}$.*

Proof. We prove the claim by induction. For $i = 1$, it holds that $\mathrm{E}_{\sigma \sim \mathcal{D}}[X_1^{(\sigma)}] = \mathrm{E}_{\sigma \sim \mathcal{D}}[Y_1^{(\sigma)}] \cdot \lambda^{-1}(\Delta_1|\Delta_1) = \frac{|A_1|}{s} \cdot \lambda(\Delta_1|\Delta_1) \cdot \lambda^{-1}(\Delta_1|\Delta_1) = \frac{|A_1|}{s}$. Assuming that the claim holds for $i - 1$, and we prove it holds for i as well. By definition, we have that

$$\mathrm{E}_{\sigma \sim \mathcal{D}}[X_i^{(\sigma)}] = \mathrm{E}_{\sigma \sim \mathcal{D}}[(Y_i^{(\sigma)} - \sum_{j \in \mathcal{G}(i)} X_j^{(\sigma)} \cdot \lambda(\Delta_i|\Delta_j)) \cdot \lambda^{-1}(\Delta_i|\Delta_i)]$$

$$= \Big(\sum_{j \in \mathcal{G}(i) \cup \{i\}} \frac{|A_j|}{s} \cdot \lambda(\Delta_i|\Delta_j) - \sum_{j \in \mathcal{G}(i)} \mathrm{E}_{\sigma \sim \mathcal{D}}[X_j^{(\sigma)}] \cdot \lambda(\Delta_i|\Delta_j) \Big) \cdot \lambda^{-1}(\Delta_i|\Delta_i)$$

$$= \Big(\sum_{j \in \mathcal{G}(i) \cup \{i\}} \frac{|A_j|}{s} \cdot \lambda(\Delta_i|\Delta_j) - \sum_{j \in \mathcal{G}(i)} \frac{|A_j|}{s} \cdot \lambda(\Delta_i|\Delta_j) \Big) \cdot \lambda^{-1}(\Delta_i|\Delta_i) = \frac{|A_i|}{s},$$

where the second to last equation follows from the induction. ◀

We can now bound the variance of $Y_i^{(\sigma)}$ as shown in the following claim.

▶ **Claim 8.** *For any $i \leq N$, it holds that $Var_{\sigma\sim\mathcal{D}}[Y_i^{(\sigma)}] \leq \frac{1}{s^2} \cdot d^{2k+1} \sum_{j\in\mathcal{G}(i)\cup\{i\}} |A_j| \cdot \lambda(\Delta_i|\Delta_j)$.*

Proof. Recall that $Y_i^{(\sigma)} = \frac{1}{s}\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}$. Note that for each $v \in A$, by the independence assumption on $\mathcal{D}$, the random variable $Z_{v,i}$ can only correlate with the corresponding variables for vertices that are within distance at most $2k$ from v. The number of such vertices is at most $1 + d + d^2 + \cdots + d^{2k} < d^{2k+1}$. Let $\mathrm{dt}(u, v)$ denote the distance between u, v in the graph G. Then we have that

$$\mathrm{E}_{\sigma\sim\mathcal{D}}[(\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i})^2] = \mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}}\sum_{\substack{u\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}\cdot Z_{u,i}]$$

$$= \mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}}(\sum_{\substack{u\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}\\ \mathrm{dt}_G(u,v)\leq 2k}} Z_{v,i}\cdot Z_{u,i} + \sum_{\substack{u\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}\\ \mathrm{dt}_G(u,v)>2k}} Z_{v,i}\cdot Z_{u,i})]$$

$$\leq \mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}}\sum_{\substack{u\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}\\ \mathrm{dt}_G(u,v)\leq 2k}} Z_{v,i}] + \left(\sum_{j\in\mathcal{G}(i)\cup\{i\}}[|A_j|]\cdot\lambda(\Delta_i|\Delta_j)\right)^2$$

$$\leq d^{2k+1}\mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}] + (\mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}])^2$$

$$= d^{2k+1}\cdot\sum_{j\in\mathcal{G}(i)\cup\{i\}}|A_j|\cdot\lambda(\Delta_i|\Delta_j) + (\mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}])^2,$$

where the first inequality follows from the fact that $Z_{u,i} \leq 1$, and that for any two vertices u, v with $\mathrm{dt}(u, v) > 2k$, $Z_{u,i}, Z_{v,i}$ are independent.

Then we have that

$$\mathrm{Var}_{\sigma\sim\mathcal{D}}[Y_i^{(\sigma)}] = \frac{1}{s^2}\cdot\mathrm{Var}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}]$$

$$= \frac{1}{s^2}\left(\mathrm{E}_{\sigma\sim\mathcal{D}}[(\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i})^2] - (\mathrm{E}_{\sigma\sim\mathcal{D}}[\sum_{\substack{v\in A_j\\ j\in\mathcal{G}(i)\cup\{i\}}} Z_{v,i}])^2\right)$$

$$\leq \frac{1}{s^2}\cdot d^{2k+1}\sum_{j\in\mathcal{G}(i)\cup\{i\}}|A_j|\cdot\lambda(\Delta_i|\Delta_j).$$ ◀

We next prove that each $X_i^{(\sigma)}$ is concentrated around its expectation with high probability.

▶ **Claim 9.** *For any $i \leq N$, it holds that $\Pr_{\sigma\sim\mathcal{D}}[|X_i^{(\sigma)} - E_{\sigma\sim\mathcal{D}}[X_i^{(\sigma)}]| \leq \theta_i\delta] \geq 1 - \beta_i$.*

Proof. We prove the claim by induction. For $i = 1$, it holds that

$$\Pr_{\sigma\sim\mathcal{D}}[|X_1^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[X_1^{(\sigma)}]| \leq \theta_1\delta] \leq \Pr_{\sigma\sim\mathcal{D}}[|Y_1^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[Y_1^{(\sigma)}]|\cdot\lambda^{-1}(\Delta_1|\Delta_1) \geq \delta\theta_1]$$

$$\leq \frac{\mathrm{Var}_{\sigma\sim\mathcal{D}}[Y_1^{(\sigma)}]}{(\delta\theta_1)^2\cdot\lambda^2(\Delta_1|\Delta_1)} \leq \frac{d^{2k+1}|A_1|\cdot\lambda(\Delta_1|\Delta_1)}{s^2\cdot(\delta\theta_1)^2\cdot\lambda^2(\Delta_1|\Delta_1)} \leq \frac{d^{2k+1}}{s(\delta\theta_1)^2\cdot\lambda(\Delta_1|\Delta_1)} \leq \beta_1,$$

where the last inequality follows from our choice of β_1, θ_1 and s which satisfy that $s \geq \frac{d^{2k+1}}{(\delta\theta_1)^2\beta_1\cdot\lambda(\Delta_1|\Delta_1)}$. Now let us consider arbitrary $i \geq 2$, assuming that the claim holds for any $j \leq i-1$. First, with probability (over the randomness that $\sigma \sim \mathcal{D}$) at least $1 - \sum_{j=1}^{i-1}\beta_j = 1 - \sum_{j=1}^{i-1} 3^{j-N-2} \geq 1 - \frac{\beta_i}{2}$, it holds that for all $j \leq i-1$, $|X_j^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[X_j^{(\sigma)}]| \leq \theta_j\delta$. This further implies that with probability at least $1 - \frac{\beta_i}{2}$,

$$\begin{aligned}
& |\sum_{j\in\mathcal{G}(i)} X_j^{(\sigma)} \cdot \frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[(\sum_{j\in\mathcal{G}(i)} X_j^{(\sigma)} \cdot \frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)})]| \\
\leq & \sum_{j\in\mathcal{G}(i)} |X_j^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[X_j^{(\sigma)}]| \cdot \frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)} \\
\leq & \sum_{j\in\mathcal{G}(i)} \delta\theta_j \cdot \frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)} \leq \kappa \cdot \sum_{j\in\mathcal{G}(i)} \delta\theta_j \leq \kappa \cdot \sum_{j=1}^{i-1} \delta(3\kappa)^{j-N} \leq \frac{\theta_i\delta}{2}.
\end{aligned}$$

Now note that

$$\begin{aligned}
& \Pr_{\sigma\sim\mathcal{U}}[|Y_i^{(\sigma)} - \mathrm{E}[Y_i^{(\sigma)}]| \cdot \lambda(\Delta_i|\Delta_i)^{-1} \geq \frac{\theta_i\delta}{2}] \leq \frac{4\cdot \mathrm{Var}_{\sigma\sim\mathcal{D}}[Y_i^{(\sigma)}]}{(\delta\theta_i)^2\cdot\lambda(\Delta_i|\Delta_i)^2} \\
\leq & \frac{4\cdot d^{2k+1}\sum_{j\in\mathcal{G}(i)\cup\{i\}}|A_j|\cdot\lambda(\Delta_i|\Delta_j)}{s^2\cdot(\delta\theta_i)^2\cdot\lambda(\Delta_i|\Delta_i)^2} \leq \frac{4\cdot d^{2k+1}\cdot\kappa}{s\cdot(\delta\theta_i)^2\cdot\lambda(\Delta_i|\Delta_i)} \leq \frac{\beta_i}{2},
\end{aligned}$$

where the last inequality follows from our choice of β_i, θ_i and s which satisfy that $s \geq \frac{8\kappa\cdot d^{2k+1}}{(\delta\theta_i)^2\beta_i\cdot\lambda(\Delta_i|\Delta_i))}$.

Therefore, with probability (over $\sigma \sim \mathcal{D}$) at least $1 - \frac{\beta_i}{2} - \frac{\beta_i}{2} = 1 - \beta_i$, it holds that

$$\begin{aligned}
& |X_i^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[X_i^{(\sigma)}]| \\
= & \left| \frac{Y_i^{(\sigma)} - \sum_{j\in\mathcal{G}(i)} X_j^{(\sigma)}\cdot\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)} - \mathrm{E}_{\sigma\sim\mathcal{D}}\left[\frac{Y_i^{(\sigma)} - \sum_{j\in\mathcal{G}(i)} X_j^{(\sigma)}\cdot\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)}\right]\right| \\
= & \left| \frac{(Y_i^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[Y_i^{(\sigma)}])}{\lambda(\Delta_i|\Delta_i)} - \left(\sum_{j\in\mathcal{G}(i)} X_j^{(\sigma)}\cdot\frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[(\sum_{j\in\mathcal{G}(i)} X_j^{(\sigma)}\cdot\frac{\lambda(\Delta_i|\Delta_j)}{\lambda(\Delta_i|\Delta_i)})]\right)\right| \\
\leq & \frac{\delta\theta_i}{2} + \frac{\delta\theta_i}{2} = \delta\theta_i.
\end{aligned}$$

◀

Now with probability (over both $A \sim \mathcal{U}_V$ and $\sigma \sim \mathcal{D}$) at least $1 - \beta_i - \beta_i$, it holds that

$$\begin{aligned}
& \left|X_i^{(\sigma)} - \frac{|V_i|}{n}\right| \leq \left|X_i^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[X_i^{(\sigma)}]\right| + \left|\mathrm{E}_{\sigma\sim\mathcal{D}}[X_i^{(\sigma)}] - \frac{|V_i|}{n}\right| \\
= & \left|X_i^{(\sigma)} - \mathrm{E}_{\sigma\sim\mathcal{D}}[X_i^{(\sigma)}]\right| + \left|\frac{|A_i|}{s} - \frac{|V_i|}{n}\right| \leq \delta\theta_i + \delta\theta_i = 2\delta\theta_i.
\end{aligned}$$

Finally, with probability at least $1 - 2\sum_{j=1}^{N}\beta_j = 1 - 2\sum_{j=1}^{N} 3^{j-N-2} \geq 1 - \frac{1}{3}$, it holds that for all $i \leq N$, $|X_i - \frac{|V_i|}{n}| \leq 2\theta_i\delta \leq \delta$. This completes the proof of the lemma. ◀

4 Constant-Query Property Testing

In this section, we show how to transform constant-query property testers in the adjacency list model to constant-space property testers in the random order stream model in a single pass and prove our main result Theorem 1. (Our transformation also works in the locally random order model as defined in Definition 3, but for simplicity, we only state our result in the uniformly random order model.)

▶ **Definition 10.** Let $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ be a property of d-bounded graphs, where Π_n is a property of graphs with n vertices. We say that Π is testable with query complexity q, if for every ε, d and n, there exists an algorithm that performs $q = q(n, \varepsilon, d)$ queries to the adjacency list of the graph, and with probability at least $2/3$, accepts any n-vertex d-bounded graph G satisfying Π, and rejects any n-vertex d-bounded graph that is ε-far from satisfying Π. If $q = q(\varepsilon, d)$ is a function independent of n, then we call Π constant-query testable.

Similarly, we can define constant-space testable properties in graph streams.

▶ **Definition 11.** Let $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ be a property of d-bounded graphs, where Π_n is a property of graphs with n vertices. We say that Π is testable with space complexity q, if for every ε, d and n, there exists an algorithm that performs a single pass over an edge stream of an n-vertex d-bounded graph G, uses $q = q(n, \varepsilon, d)$ space, and with probability at least $2/3$, accepts G if it satisfies Π, and rejects G if it is ε-far from satisfying Π. If $q = q(\varepsilon, d)$ is a function independent of n, then we call Π constant-space testable.

The proof of Theorem 1 is based on the following known fact: every constant-query property tester can be simulated by some canonical tester which only samples a constant number of vertices, and explores the k-discs of these vertices, and then makes deterministic decisions based on the explored subgraph. This implies that it suffices to approximate the distribution of k-disc types of the input graph to test the corresponding property. Formally, we will use the following lemma relating the constant-time testable properties and their k-disc distributions. For any graph G, let $S_{G,k}$ denote the subgraph spanned by the union of k-discs rooted at k uniformly sampled vertices from G. The following lemma is implied by Lemma 3.2 in [7] (which was built on [14] and [13]). (The result in [7] is stated for d-bounded directed graphs, while it also holds in the undirected case.)

▶ **Lemma 12.** *Let $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ be any d-bounded graph property that is testable with $q = q(\varepsilon, d)$ query complexity in the adjacency list model. Then there exist integer n_0, $k = c \cdot q$ for some large universal constant c, and an infinite sequence of $\mathcal{F} = \{\mathcal{F}_n\}_{n\geq n_0}$ such that for any $n \geq n_0$, $\mathcal{F}_n$ is a set of digraphs, each being a union of k disjoint k-discs, and for any n-vertex graph G,*

- *if G satisfies Π_n, then with probability* at most $\frac{5}{12}$, *$S_{G,k}$ is isomorphic to one of the members in $\mathcal{F}_n$.*
- *if G is ε-far from satisfying Π_n, then with probability* at least $\frac{7}{12}$, *$S_{G,k}$ is isomorphic to one of the members in $\mathcal{F}_n$.*

Now we are ready to prove Theorem 1.

Proof Sketch of Theorem 1. The proof follows almost directly from the proof of Theorem 1.1 in [7]. We sketch the algorithm and its analysis, and refer to [7] for further details.

The algorithm is as follows. For any property $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ that is testable with query complexity $q = q(\varepsilon, d)$ in the adjacency list model, we set $k = c \cdot q$ as guaranteed in Lemma 12, and set $N = N(d, k)$, $\delta = \frac{1}{48(2kN)^k}$. Let G be any n-vertex graph with $n \geq n_1 := n_1(d, k)$ that is represented by a uniformly random order edge stream, where n_1 is some sufficiently large constant. (For graphs with $n < n_1$ vertices, one can trivally test Π_n with constant space.) Let Λ_k be the set of probabilities as guaranteed in Lemma 4. We first invoke the algorithm k-Disc_Distribution(Stream(G), $\Lambda_k, n, d, k, \delta$) to get estimators $X_1, \cdots, X_N$ for the fraction $f_1, \cdots, f_N$ of vertices whose k-discs are isomorphic to $\Delta_1, \cdots, \Delta_N$, respectively. As guaranteed by Lemma 5, with probability at least $2/3$, it holds that for any $i \leq N$, $|X_i - f_i| \leq \delta$. Conditioned on this event, we approximate the frequency of each subgraph F in $\mathcal{F}_n$ as guaranteed in Lemma 12, where $F = (\Gamma_1, \cdots, \Gamma_k)$ is a multiset of k-discs. That is, for each $F = (\Gamma_1, \cdots, \Gamma_k)$, we calculate its empirical frequency as $\mathrm{estim}(F) = \frac{\prod_{i=1}^{N} \binom{X_i \cdot n}{x_i}}{\binom{n}{k}}$, where x_i is the number of copies among $\Gamma_1, \cdots, \Gamma_k$ that are of the same type as Δ_i, for $1 \leq i \leq N$. Finally, we accept the graph if and only if $\sum_{F \in \mathcal{F}_n} \mathrm{estim}(F) < \frac{1}{2}$.

By Lemma 5 and our setting of δ, the space used by the algorithm is $O(\frac{\kappa^{2N} \cdot d^{3k+2} \cdot 3^{3N+1}}{\delta^2 \lambda_{\min}}) = O(\frac{\kappa^{2N} \cdot d^{3k+2} \cdot 3^{3N+1} \cdot (2kN)^{2k}}{\lambda_{\min}})$.

For the correctness of the algorithm, note that if X_i's are good estimators for f_i's, then the empirical probability $\mathrm{estim}(F)$ is close to the probability that $S_{G,k}$, the subgraph spanned by the union of k-discs rooted at k uniformly sampled vertices from G, spans a subgraph that is isomorphic to F. This implies that the quantity $\sum_{F \in \mathcal{F}_n} \mathrm{estim}(F)$ is a good estimator for the probability that $S_{G,k}$ is isomorphic to one of the members in $\mathcal{F}$. Combining this with Lemma 12, we can show that if G satisfies Π_n, then $\sum_{F \in \mathcal{F}_n} \mathrm{estim}(F) < \frac{1}{2}$ and if G is ε-far from satisfying the property Π_n, then $\sum_{F \in \mathcal{F}_n} \mathrm{estim}(F) \geq \frac{1}{2}$. We omit details here. ◀

5 Constant-Time Approximation Algorithms

As we mentioned in the introduction, to simulate any constant-time algorithm that is independent of the labeling of the vertices, and accesses the graph by sampling random vertices and exploring neighborhoods (or k-discs for some k) of these vertices, it suffices to have the distribution of k-disc types. Now we explain slightly more about this simulation and sketch the proof of Theorem 2. In order to approximate the size of the solution of an optimization problem (e.g., maximum matching, minimum vertex cover), it has been observed by Parnas and Ron [28] that it suffices to have efficient oracle $\mathcal{O}_S$ access to a solution S. This is true since one can attain a good estimator for the size of S by sampling a constant number of vertices, performing corresponding queries to the oracle $\mathcal{O}_S$ and then returning the fraction of vertices that belong to S based on the returned answers from $\mathcal{O}_S$. Nguyen and Onak [26] implemented such an oracle via an elegant approach of locally simulating the classical greedy algorithm. In particular, they showed the following result.

▶ **Lemma 13** ([26]). *There exist $q = q(\varepsilon, d)$, an oracle $\mathcal{O}_M$ to a maximal matching M, and an algorithm that queries $\mathcal{O}_M$ about all the edges incident to a set of $s = O(1/\varepsilon^2)$ randomly sampled vertices and with probability at least $2/3$, returns an estimator that is $(1, \varepsilon n)$-approximation of the size of M, and each query to $\mathcal{O}_M$ performs at most q queries to the adjacency list of the graph.*

A key observation is that the algorithm in Lemma 13 can be viewed as first sampling s q-discs from the graph and then perform $\mathcal{O}_M$ queries on each of these q-discs. It is easy to see that with high probability 0.99, all these q-discs are disjoint. Furthermore, the answer of the above oracle only depends on the structure of the corresponding neighborhood of the starting vertex v and the random ordering of the edges belonging to this neighborhood. Now we can approximate the size of a maximal matching in the random order streaming model as follows: we first invoke Algorithm 2 to get an estimator for the distribution of q-discs. Then we can simulate the oracle on this distribution.

Acknowledgments. We would like to thank G. Cormode, H. Jowhari for helpful discussions.

References

1 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.

2 Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1723–1742. SIAM, 2017.

3 Itai Benjamini, Oded Schramm, and Asaf Shapira. Every minor-closed property of sparse graphs is testable. *Advances in mathematics*, 223(6):2200–2218, 2010.

4 Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms-ESA 2015*, pages 263–274. Springer, 2015.

5 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 641–650. ACM, 2008.

6 Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM J. Comput.*, 34(6):1370–1379, 2005.

7 Artur Czumaj, Pan Peng, and Christian Sohler. Relating two property testing models for bounded degree directed graphs. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 1033–1045. ACM, 2016.

8 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theoretical Computer Science*, 348(2-3):207–216, 2005.

9 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM Journal on Computing*, 38(5):1709–1727, 2008.

10 Joan Feigenbaum, Sampath Kannan, Martin Strauss, and Mahesh Viswanathan. Testing and spot-checking of data streams. *Algorithmica*, 34(1):67–80, 2002.

11 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

12 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32:302–343, 2002.

13 Oded Goldreich and Dana Ron. On proximity-oblivious testing. *SIAM Journal on Computing*, 40(2):534–566, 2011.

14 Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Structures & Algorithms*, 23(1):23–57, 2003.

15 Avinatan Hassidim, Jonathan A Kelner, Huy N Nguyen, and Krzysztof Onak. Local graph partitions for approximation and testing. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 22–31. IEEE, 2009.

16 Monika Rauch Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. In *External Memory Algorithms, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, May 20-22, 1998*, pages 107–118, 1998.

17 Zengfeng Huang and Pan Peng. Dynamic graph stream algorithms in o(n) space. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 18:1–18:16, 2016.

18 Hiro Ito, Shin-Ichi Tanigawa, and Yuichi Yoshida. Constant-time algorithms for sparsity matroids. In *International Colloquium on Automata, Languages, and Programming*, pages 498–509. Springer, 2012.

19 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 734–751. Society for Industrial and Applied Mathematics, 2014.

20 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating max-cut. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1263–1282. Society for Industrial and Applied Mathematics, 2015.

21 Ken-ichi Kawarabayashi and Yuichi Yoshida. Testing subdivision-freeness: property testing meets structural graph theory. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 437–446. ACM, 2013.

22 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 231–242. Springer, 2012.

23 Sharon Marko and Dana Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Transactions on Algorithms (TALG)*, 5(2):22, 2009.

24 Andrew McGregor. Graph stream algorithms: a survey. *ACM SIGMOD Record*, 43(1):9–20, 2014.

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

26 Huy N Nguyen and Krzysztof Onak. Constant-time approximation algorithms via local improvements. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 327–336. IEEE, 2008.

27 Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1123–1131. SIAM, 2012.

28 Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theoretical Computer Science*, 381(1):183–196, 2007.

29 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

30 Xiaoming Sun and David P Woodruff. Tight bounds for graph problems in insertion streams. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 40. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

31 Shin-Ichi Tanigawa and Yuichi Yoshida. Testing the supermodular-cut condition. *Algorithmica*, 71(4):1065–1075, 2015.

32 Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 11–25. SIAM, 2011.

33 Yuichi Yoshida and Hiro Ito. Property testing on k-vertex-connectivity of graphs. In *Automata, Languages and Programming*, pages 539–550. Springer, 2008.

34 Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. Improved constant-time approximation algorithms for maximum matchings and other optimization problems. *SIAM J. Comput.*, 41(4):1074–1093, 2012.