Testable Properties in General Graphs and Random Order Streaming

Artur Czumaj
Department of Computer Science and Centre for Discrete Mathematics and its Applications (DIMAP), University of Warwick, Coventry, UK
A.Czumaj@warwick.ac.uk

Hendrik Fichtenberger
Department of Computer Science, TU Dortmund, Germany
hendrik.fichtenberger@tu-dortmund.de

Pan Peng
Department of Computer Science, University of Sheffield, UK
p.peng@sheffield.ac.uk

Christian Sohler
Department of Mathematics and Computer Science, University of Cologne, Germany
csohler@uni-koeln.de

Abstract

We consider the fundamental question of understanding the relative power of two important computational models: *property testing* and *data streaming*. We present a novel framework closely linking these areas in the setting of *general graphs* in the context of constant-query complexity testing and constant-space streaming. Our main result is a generic *transformation* of a one-sided error property tester in the random-neighbor model with constant *query* complexity into a one-sided error property tester in the streaming model with constant *space* complexity. Previously such a generic transformation was only known for *bounded-degree* graphs.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Graph property testing, sublinear algorithms, graph streaming algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2020.16

Category RANDOM

Related Version The full version of the paper is available at `https://arxiv.org/abs/1905.01644`. All the missing proofs can be found in the full version.

Funding *Artur Czumaj*: Research partially supported by the Centre for Discrete Mathematics and its Applications (DIMAP), by IBM Faculty Award, and by EPSRC award EP/N011163/1.
Hendrik Fichtenberger: Research supported by ERC grant No. 307696.
Christian Sohler: Research supported by ERC grant No. 307696.

Acknowledgements We would like to thank anonymous reviewers for extensive comments.

1 Introduction

We consider the fundamental question of understanding the relative power of two important computational models: *property testing* and *data streaming*. We present a novel framework closely linking these areas in the setting of *general graphs* in the context of constant-query complexity testing and constant-space streaming. We first provide a new analysis of constant-query property testers (in the random-neighbor model, see Definition 6) for general graphs

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020).
Editors: Jarosław Byrka and Raghu Meka; Article No. 16; pp. 16:1–16:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and develop the framework of canonical testers for general graphs. Then, using the concept of canonical testers, we provide a generic *transformation* of a one-sided error property tester in the random-neighbor model with constant *query* complexity into a one-sided error property tester in the streaming model with constant *space* complexity.

Property testing. A fundamental task in the study of big networks/graphs is to efficiently analyze their structural properties. For example, we may want to know if a graph is well-connected, has many natural clusters, has many copies (instances) of some specific sub-structures, etc. Given that modern networks are large, often consisting of millions and billions of nodes (web graph, social networks, etc.), the task of analyzing their structure has become recently more and more challenging, and the running-time efficiency of this task is becoming of critical importance. The framework of *property testing* has been developed to address some of these challenges, aiming to trade the efficiency with the accuracy of the output, with the goal of achieving very fast algorithms.

In (graph) property testing, one of the main challenges is to characterize properties that are testable with a constant number of queries in various computational models. Typically, a tester has query access to a graph (e.g., random vertices or neighbors of a vertex for graphs), and its goal is to determine if the graph satisfies a certain property (e.g., is well-clusterable) or is far from having such a property (e.g., is "far" from any graph being well-clusterable; see, e.g., [18, 19, 20, 39]). To be precise, we define testers as follows. Given a property Π, a tester for Π is a (possibly randomized) algorithm that is given a proximity parameter ε and oracle access to the input graph G. If G satisfies property Π, then the algorithm must accept with probability at least $\frac{2}{3}$. If G is ε-far from Π, then the algorithm must reject with probability at least $\frac{2}{3}$. If the algorithm is allowed to make an error in both cases, we say it is a *two-sided error tester*; if, on the contrary, the algorithm always gives the correct answer when G satisfies the property, we say it is a *one-sided error tester*. Further details of the model depend on the data representation. In the main model considered in this paper, *property testing for general graphs*, we will consider the *random neighbor oracle access* to the input graph (cf. Definition 6), which allows to query a random neighbor of any given vertex[1]. In our model, we will say that G is ε-far from a property Π if any graph that satisfies Π differs from G on at least $\varepsilon|E(G)|$ edges. To analyze the performance of a tester, we will measure its quality in term of its *query complexity*, which is the number of oracle queries it makes.

In the past, a large body of research has focused on the analysis of various graph properties in different graph models, for example, leading to a precise characterization of all properties that can be tested with constant query complexity [1, 3] in the so-called dense model (graphs with $\Theta(n^2)$ edges), and some partial results for bounded-degree graph models (see, e.g., [5, 13, 17, 18, 20, 21, 35]). However, our understanding of the model of general graphs, graphs where each vertex can have arbitrary degree, is still rather limited. We have seen some major advances in testing graph properties for general graphs, including the results of Parnas and

[1] Our model is in contrast with the other two widely used property testing models for graphs with arbitrarily large maximum degree: In the *adjacency list model* [36, 30], the algorithm can perform both *neighbor queries* (i.e., for the i-th neighbor of any vertex v such that $i \leq \deg(v)$), and the *degree queries* (i.e., for the degree $\deg(v)$ of any vertex v); In the *general graph model*, the algorithm is allowed to perform *vertex-pair queries* (i.e., for the existence of an edge between any two vertex pair u, v), in addition to neighbor and degree queries [28, 2, 18]. Still, we believe that the *random neighbor oracle model* considered in this paper is the most natural model of computations in the property testing framework in the context of very fast algorithms, especially those performing $O(1)$ queries. We note however, that our analysis can be generalized to other models of general graphs (cf. the full version).

Ron [36], Kaufman et al. [28], Alon et al. [2], Czumaj et al. [11, 14] (see also the survey in [18, Chapter 10]). The main challenge of the study in the model of general graphs is a lack of good characterization of testable properties and of a good algorithmic toolbox for the problems in this model. Still, the importance of the general graph model and lack of major advances have been widely acknowledged in the property testing community. For example, it is recognized that the general graph model is "most relevant to computer science applications" and "designing testers in this model requires the development of algorithmic techniques that may be applicable also in other areas of algorithmic research" (see [18, Chapter 10.5.3]).

Graph streaming algorithms. One important way of processing large graphs in modern data analysis is to design *graph streaming algorithms* (see, e.g., [31, 34]). A graph streaming algorithm obtains the input graph as a stream of edges in some order and its goal is to process and analyze the input stream in order to compute some basic characteristics about the input graph. For example, we want to know whether the graph is connected, or bipartite, or to know its approximate maximum matching size. Following the mainstream research in data streaming, we focus on algorithms that make only a *single pass* over the graph stream. Since in the single pass model every edge is seen only once, the central complexity measure of data streaming algorithms is the amount of space used to store information about the graph, with the golden standard in streaming being *sublinear space*. Unfortunately, it is known that for many natural graph problems sublinear space $o(n)$ is not possible when the edges are arriving in a single pass and in arbitrary order, where n is the number of vertices of the input graph [23].

There have been several approaches to cope with this inherent limitation of the streaming setting for graph problems. While some of the early works in graph streaming algorithms approached this challenge by allowing more than one pass over the input, the single-pass model is still considered to be the most interesting and the most natural scenario for streaming algorithms. The $\Omega(n)$ space lower bound (e.g., for testing if the graph is connected or estimating the size of transitive closure [23]) led to a significant number of papers designing semi-streaming algorithms, which are algorithms using $O(n \text{ polylog}(n))$ space, so only slightly larger than linear in the number of vertices (see the survey [31]). This model leads to sublinear algorithms for dense graphs, where m, the number of edges, is $\omega(n \text{ polylog}(n))$. For the very natural setting of *sparse graphs*, semi-streaming algorithms are useless, since with $O(n \text{ polylog}(n))$ space one can store the entire input graph (all arriving edges). Therefore, one can trivially solve any graph problem. Some works consider special classes of graphs. For example, it is known how to approximate the matching size within a constant factor in polylogarithmic space for planar graphs or graphs with bounded arboricity (see, e.g., [15, 10, 32, 6]).

Another, central approach to address the linear space lower bounds for graph streaming problems that recently received increasing attention is the *random-order streaming model*, where the edges arrive in random order, i.e., in the order of a uniformly random permutation of the edges (see, e.g., [8, 26, 29, 31, 33, 37, 4, 27, 9, 16]). The assumption about uniformly random or near-uniformly random ordering is very natural and can arise in many contexts. One might also use the random-order streaming model to justify the success of some heuristics in practice, even though there exists strong space lower bound for (the worst case of) the problem. Furthermore, some recent advances have shown that some problems that are hard for adversarial streams can be solved with small space in the random order model. For example, Konrad et al. [29] gave single-pass semi-streaming algorithms for maximum matching for bipartite and general graphs with approximation ratio strictly larger than $\frac{1}{2}$ in the random order semi-streaming model. Kapralov et al. [26] 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. It is not known if such trade-offs between approximation ratios and space complexity are possible in the adversarial order model. Finally, [37] showed that in the random-order streaming model, even with constant space, one can approximate the number of connected components of the input graph within an additive error of εn, the size of a maximum independent set in planar graphs within a multiplicative factor of $1+\varepsilon$, and the weight of a minimum spanning tree of a connected input graph with small integer edge weights within a multiplicative factor of $1+\varepsilon$. However, for the first and third problems in adversarial order streams, there exist $n^{1-O(\varepsilon)}$ space lower bounds [24]. While these results demonstrate the strength of the random-order streaming model, Chakrabarti et al. [8] proved that $\Omega(n)$ space is needed for any single pass algorithm for graph connectivity in the random-order streaming model, almost matching the optimal $\Omega(n \log n)$ space lower bound in the adversarial order model [40]. This poses a central open question in the area of graph streaming algorithms, of *characterizing graph problems which can be solved with small, sublinear space in the random-order streaming model.*

The main goal of our paper in the context of streaming algorithms, is to address this task and to enlarge the class of graph problems known to be solvable with *small space* in the *random order streaming* model in a *single pass*. Our main focus is on the most challenging scenario: of achieving *constant space*[2].

1.1 Basic Definitions and Overview of Our Results

In this paper, we extend the approach recently introduced by Monemizadeh et al. [33] (see also [37]) to demonstrate a *close connection between streaming algorithms and property testing* in the most general setting of *general graphs*. Monemizadeh et al. [33] show that for *bounded-degree graphs*, any graph 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. In this paper, we show that similar results hold also for general graphs. To this end, we design a novel framework of canonical testers for all constant-query testers for general graphs and apply it to design a generic method of transforming any constant-query tester (with one-sided error) for graph properties into a constant-space tester (with one-sided error) in the random-order streaming model.

We consider the *random neighbor* query oracle model for general graphs, which allows the algorithm to query a random neighbor of any specified vertex (cf. Definition 6).

▶ **Definition 1** (Property testers in the random-neighbor model). *Let $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ be a graph property, where Π_n is a property of graph of n vertices. We say that Π* is testable with query complexity *q, if for every ε and n, there exists an algorithm (called* tester*) that makes at most $q = q(n, \varepsilon)$ oracle queries, and with probability at least $\frac{2}{3}$, accepts any n-vertex graph satisfying Π, and rejects any n-vertex graph that is ε-far from satisfying Π. If $q = q(\varepsilon)$ is a function independent of n, then we call Π* constant-query testable*. If the tester always accepts graphs that satisfy Π, we say that it has* one-sided error*. Otherwise, we say the tester has* two-sided error*.*

[2] Throughout the entire paper, we will count the size of the *space in words* (assuming that a single word can store any single ID of a vertex or of an edge), i.e., space bounds have to be multiplied by $O(\log n)$ to obtain the number of bits used. With this in mind, we use term *constant space* to denote space required to store a constant number of words, or IDs, that is, $O(\log n)$ bits.

We notice that the definition above is generic and can be applied to any of the query oracle models (see e.g. [18]). However, since our main query oracle model is the random-neighbor model, only for that model we will use the terminology from Definition 1 without a direct reference to the query oracle model. We first present *canonical testers* in this model. In order to do so, we introduce a process called *q-random BFS* (q-RBFS) starting with any specified vertex v, i.e., a BFS of depth q that is restricted to visiting at most q random neighbors for every vertex (see Definition 7). We call the subgraph obtained by a q-RBFS a *q-bounded disc*. Our first result is informally stated as follows.

▶ **Theorem 2** (informal; cf. Theorem 10). *If a property $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ is testable with $q = q(\varepsilon)$ queries in the random-neighbor model, then it can also be tested by a canonical tester that*

1. *samples q' vertices;*
2. *performs q'-RBFS from each sampled vertex;*
3. *accepts if and only if the explored subgraph does not contain any (forbidden) graph $F \in \mathcal{F}$,*

where q' depends only on q, and $\mathcal{F}$ is a family of rooted graphs such that each graph $F \in \mathcal{F}$ is the union of q' many q'-bounded discs.

We remark that similar canonical testers have been given for dense graphs [22], bounded-degree graphs and digraphs [21, 12]. Actually, our proof for the above theorem heavily builds upon [21, 12], though our analysis requires some extensions to deal with general graphs (of possibly unbounded degree). To formally state our result regarding testing graph properties in streaming, we introduce the following definition.

▶ **Definition 3** (Property testers in the streaming model). *Let $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ be a graph property, where Π_n is a property of graph of n vertices. We say that Π is testable with space complexity q, if for every ε and n, there exists an algorithm that performs a single pass over an edge stream of an n-vertex graph G, uses $q = q(n, \varepsilon)$* words *of space, and with probability at least $\frac{2}{3}$, accepts G satisfying Π, and rejects G that is ε-far from satisfying Π. If $q = q(\varepsilon)$ is a function independent of n, then we call Π* constant-space testable*. If the tester always accepts the property, then we say that the property can be tested with* one-sided error*. Otherwise, we say the tester has* two-sided error*.*

Our main result and our main technical contribution is the *transformation* of a one-sided error property tester in the random-neighbor model with constant *query* complexity into a one-sided error property tester in the streaming model with constant *space* complexity.

▶ **Theorem 4** (Main Theorem). *Every graph property Π that is constant-query testable with one-sided error in the random-neighbor model is also constant-space testable (space measured in words) with one-sided error in the random order graph streams.*

Applications. We believe that the main contribution of our paper is the general transformation presented in Theorem 4. However, we admit that the number of properties testable with one-sided error with a constant number of queries in the random-neighbor model is rather limited. Still, we can apply our transformation to, for example, the property of being (s, t)-disconnected (i.e., there is no path between s and t), see, e.g., [41][3]. Furthermore, our

[3] The constant-query tester from [41] performs degree queries and neighbor queries, but it is straightforward to simulate it in the random-neighbor model. Indeed, the algorithm in [41] only needs to repeatedly perform a constant-length random walks from s and reject if only if one path from s to t is found. Such an algorithm can be trivially simulated in the random-neighbor model, as each step of a random walk just needs to query one random neighbor of the current vertex.

transformation actually holds when the input graph is restricted to come from a certain class of graphs such as planar graphs, minor-free graphs, or bounded-degree graphs. Since bipartiteness in planar graphs (or minor-free graphs) is testable in the random-neighbor model [11], it is also one-sided error testable in random order streams in constant space; notice that this result stands in contrast to the $n^{1-O(\varepsilon)}$ space lower bound for *adversarial order streams* for (property) testing bipartiteness in planar graphs [24]. Further, recent constant-query complexity testing of H-freeness in planar or minor-free graphs [14] shows that also testing H-freeness is one-sided error testable in random order streams in constant space.

Furthermore, our techniques can also be used to transform any constant-query tester (with one-sided error) in the *random neighbor/edge model* (cf. the full version) to the random-order streaming model, where the random neighbor/edge model allows to sample an edge uniformly at random. Therefore, for example, since the property of being P_k-free (there is no path of length k) is constant-query testable in the random neighbor/edge model with one-sided error [25], P_k-freeness is also constant-space testable with one-sided error in the random order graph streams. Similarly, it is not hard to see that the property of being d-bounded (the maximum degree is at most d) is constant-query testable in the random neighbor/edge model[4], and therefore this property too is constant-space testable with one-sided error in the random order graph streams.

The contribution of our paper goes beyond just establishing a connection between property testing and streaming. While the concept of canonical testers has been used in graph property testing before (cf. [22, 21, 12]), our study and characterization of canonical testers for general graphs (Theorem 2 and Theorem 10) is new. We believe that this study will shed light on our understanding of constant-query testable graph properties and will lead to new results for property testing in general graphs. For example, Czumaj and Sohler [14] recently used our canonical testers as a tool in their proof of a complete characterization of constant-query testable properties in general planar graphs [14] after a preliminary version of this work appeared.

1.2 Challenges and Techniques

The result about constant-space streaming algorithms for bounded-degree graphs by Monemizadeh et al. [33] is obtained by noting that any constant-query complexity tester basically estimates the distribution of local neighborhoods of the vertices (see, e.g., [12, 18, 21]) and emulating any such algorithm on a random order graph stream using constant space. Unfortunately, this approach inherently relies on the assumption that the input graph is of bounded degree. This limitation comes from two ends: on one hand, there has not been known any versatile description of testers for constant-query testable graph properties of general graphs, and on the other hand, the streaming approach from [33] relies on a breadth-first-search-like graph exploration that is possible (with constant space) only when the input graph has no high-degree vertices. A follow-up paper [37] made the first attempt to address the challenge of dealing with general degrees, and considered some problems in which one can *ignore* high degree vertices (e.g., for approximating the number of connected components or the size of a maximum independent set in planar graphs).

[4] If G is ε-far from the property, then at least $\Omega(\varepsilon|E|)$ edges are incident to a node with degree at least $d+1$. Thus, we can simply sample a constant number of edges and check if either of its endpoints has degree at least $d+1$.

One important reason why the earlier approaches have been failing for the model of general graphs, without bounded-degree assumption, was our lack of understanding of constant-query complexity testers in general graphs and the lack of techniques to appropriately emulate off-line algorithms allowing many high-degree vertices. In this paper, we advance our understanding on both of these challenges.

A general and simple canonical tester. To derive a canonical tester for constant-query testable properties in the random-neighbor model, we introduce the process *q-random BFS* (q-RBFS): it starts from any specified vertex v, and then performs a BFS-like exploration of depth q that is restricted to visiting at most q random neighbors at each step (see Definition 7 for the formal definition). We call the subgraph obtained by a q-RBFS a *q-bounded disc.* With the notion of q-RBFS and q-bounded discs, we are able to transform every constant-query tester for properties of general graphs into a *canonical tester* that works as follows: it samples q random vertices, performs a q-RBFS from each sampled vertex, and rejects if and only if the (non-induced) subgraph it has seen (which is a union of q-bounded discs) is isomorphic to some member of a family $\mathcal{F}$ of forbidden subgraphs (see Theorems 2 and 10). Furthermore, such a canonical tester preserves one-sided error, while the query complexity blows up exponentially. We believe that the exponential blow-up is necessary, even for bounded-degree graphs, as adaptivity is essential for property testing in sparse graphs [38, 7]. This is in contrast to the dense graph model for property testing, in which a quadratic blow-up of the query complexity of canonical testers was known [22].

Canonical testers provide us a systematic view of the behavior of constant-query testers in the random-neighbor model. They further tell us that in order to test a constant-query testable property Π, it suffices to estimate the probability that some forbidden subgraph in $\mathcal{F}$ is found by a q-RBFS starting from a randomly sampled vertex. Slightly more formally, we define the *reach probability* of a subgraph $F \in \mathcal{F}$ to be the probability that a q-RBFS starting from a uniformly chosen vertex v sees a graph that is isomorphic to F. In particular, if we can estimate these reach probabilities in random order streams, then we can also test Π accordingly.

The problem with this approach is that it is hard to estimate the reach probabilities of subgraphs in $\mathcal{F}$. The main challenge here is that a forbidden subgraph $F \in \mathcal{F}_n$ may be the union of more than two or more subgraphs obtained from different q-RBFS that may intersect with each other.

A refined canonical tester. To cope with the challenge mentioned above of estimating the reach probabilities of subgraphs in $\mathcal{F}$, we decompose each forbidden subgraph $F \in \mathcal{F}_n$ into all possible sets of intersecting q-bounded discs whose union is F and then try to recover F from these sets. In order to recover F from such a decomposition, we have to identify and monitor *vertices that are contained in more than one q-bounded disc of F.*

We refine the analysis of the canonical tester and separate the q-bounded discs explored by each q-RBFS and keep track of their intersections (cf. Theorem 17). We first observe that for every input graph G and every ε, there exists a *small fixed set* $V_\alpha \subseteq V$ of all vertices whose probability to be visited by a random q-RBFS from a random vertex exceeds some small threshold α (depending on q and ε, but independent of n). In other words, with constant probability, the subgraphs explored by multiple q-RBFS in the canonical tester will only overlap on vertices from V_α. Furthermore, we prove that the degree of all vertices in V_α is at least linear (in n), and with constant probability, two random q-RBFS subgraphs will not share any edge. Since V_α has constant size, each q-bounded disc can be viewed as a *colored* q-bounded disc *type* such that each vertex in V_α is assigned a unique color from a constant-size palette. This way, it is possible to reversibly decompose each $F \in \mathcal{F}_n$ into a multiset of colored q-bounded disc types (actually, there may be many such multisets for each F): since the q-bounded discs that are explored by different q-RBFS intersect only at vertices in V_α, F is obtained by identifying vertices of the same color. See Figure 1 in Appendix A for an example.

These properties are crucial to describe the forbidden subgraphs in terms of the graphs seen by the q many q-RBFS that the canonical tester performs and a *constant-size* description of their interaction, i.e., how they overlap.

Simulation in the streaming. In the streaming, in order to simulate q-RBFS, it is natural to consider the following procedure called StreamCollect (q-SC, see Algorithm 2 in Appendix B) to explore the subgraph surrounding any specified vertex. That is, it maintains a connected component C that initially contains only the start vertex. Whenever it reads an edge that connects to the current C and the augmented component may be observed by a run of q-RBFS, it adds the edge to C.

Note that one important feature of random order streams is that we would see the right exploration (as in the query model) with constant probability, while it is challenging to verify if the subgraph we collected from the stream is indeed the right exploration (cf. [33, 37] for a more detailed discussion). In our setting, as we mentioned, another technical difficulty is to analyze whether subgraphs found by running the stream procedure multiple times *intersect* in exactly the same way as the q-bounded discs that are found by q-RBFS.

With the refined canonical tester, which specifies how different q-RBFS procedures intersect, we are able to *simulate one-sided error constant-query testers* in the random-neighbor model for general graphs in the *random-order streaming model*. Since the considered property Π is one-sided error testable in the random-neighbor model, it suffices to detect a forbidden subgraph F in the family $\mathcal{F}$ corresponding to Π with constant probability. That is, it suffices to show that if the graph is far from having the property, then for any forbidden subgraph H that can be reached by the canonical tester with probability p, it can also be detected by *multiple* StreamCollect subroutines with probability at least cp for some suitable constant c.[5]

In order to do so, we first decompose the forbidden subgraphs that characterize the property into colored subgraphs, where each subgraph corresponds to a run of q-RBFS and vertices in V_α are colored with a unique color. Then, we prove that for a sufficiently large sample of vertices, the q-SC subroutines starting from these sampled vertices will collect, for each colored subgraph H, at least as many instances of H as the canonical property tester sees. Suppose that the input graph is far from the property. Since the subgraphs observed by the canonical tester intersect only at vertices in V_α, i.e., colored vertices, with constant probability, it is possible to stitch a forbidden subgraph by identifying vertices of the same color in the analysis.

[5] Note that this is not sufficient for simulating two-sided error testers. Let us take the property connectivity (which is 2-sided error testable in random-neighbor model) for example. If the input graph is a path on n vertices, then a q-RBFS will detect a forbidden subgraph (i.e., a path of constant length that is not connected to the rest) corresponding to connectivity with small constant probability, while a q-SC might see a forbidden subgraph with high constant probability. That is, in order to test connectivity, we need to be able to approximate the *frequencies* of the forbidden subgraphs, for which our current techniques fail.

The analysis of this procedure is two-fold. First, we show that if a single run of q-RBFS from v sees a certain colored q-bounded disc type with probability p (where the colored vertices are V_α), then a single run of q-SC from v sees this disc type with probability cp for some suitable constant c (see Corollary 20).

The second step (which is the main technical part) is to show that if the probability that a q-RBFS from a random vertex sees a colored q-bounded disc type Δ is p, then with constant probability, for a sufficiently large sample set S, the calls to q-SC from vertices in S will also see a q-bounded disc type Δ, even though there are intersections from different q-SCs (see Lemma 21). Then we can show that if the input graph is far from the property, with constant probability, we can stitch the colored q-bounded discs to obtain a forbidden subgraph $F \in \mathcal{F}$ (see Theorem 4).

Finally, we remark that colors are only used in the analysis as the streaming algorithm can identify intersections of multiple q-SC by the vertex labels. However, the colors are crucial to the analysis: without colors, we cannot guarantee that the q-bounded disc types found by multiple q-SCs can be stitched in the same way as the q-bounded disc types found by q-RBFS. Here is an example: Consider some constant-query testable property Π such that the set of forbidden subgraphs $\mathcal{F}$ contains a graph F that is not a subgraph of any single q-bounded disc type (i.e, it is the union of at least two intersecting q-bounded disc types). For the sake of illustration, a concrete example is provided in Figure 2 in Appendix A. In order to reject, the canonical property tester needs to find at least two intersecting q-bounded discs such that their union contains F as a subgraph. However, even if we bound, for each *uncolored* q-bounded disc type Δ, the probability that q-SC finds Δ by some constant fraction of the probability that q-RBFS finds Δ, this is not sufficient to conclude that the probability that multiple q-SCs find a copy of F is bounded by a constant fraction of the probability that multiple q-RBFS find a copy of F. The reason is that q-SC might only find copies of Δ that are not intersecting, while q-RBFS might tend to find copies of Δ that intersect. Again, see Figure 2 for an example. Therefore, we need to preserve, for each q-bounded disc type Δ, the information which of the corresponding vertices in the input graph are likely to be contained in more than one q-RBFS for the analysis.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph. We will assume that the vertex set V of G is $[n] = \{1, \ldots, n\}$, and we let $\deg(v)$ denote the degree of $v \in V$. Sometimes, we use $V(G)$ to denote the vertex set V of G and $E(G)$ to denote the edge set E of G. We let $\mathcal{S}(G)$ denote the input stream of edges that defines G. In this paper, we consider streaming algorithms for random order streams, i.e., the input stream $\mathcal{S}(G)$ to our algorithm is drawn uniformly from the set of all permutations of E. We are interested in *streaming algorithms that have constant space complexity* in the size of the graph, where we count the size of the space in words, i.e., space bounds have to be multiplied by $O(\log n)$ to obtain the number of bits used, see also Footnote 2.

A graph G is called a *rooted* graph if at least one vertex in G is marked as *root*. Let us define the notion of a root-preserving isomorphism.

▶ **Definition 5.** *Given two rooted graphs H_1 and H_2, a* root-preserving isomorphism *from H_1 to H_2 is a bijection $f : V(H_1) \to V(H_2)$ such that 1) if u is the root of $V(H_1)$ then $f(u)$ is the root of $V(H_2)$, and 2) that $(u, v) \in E(H_1)$ if and only if $(f(u), f(v)) \in E(H_2)$. If there is a root-preserving isomorphism from H_1 to H_2 then we say that H_1 is root-preserving isomorphic to H_2 and denote it by $H_1 \simeq H_2$.*

3 Canonical Constant-Query Testers in General Graphs

In this section, we present our main result on the *canonical testers for constant-query testable properties in general graphs.* After starting with some basic definitions, we will present two canonical testers for constant-query testable properties in general graphs. Our first canonical tester is of a general form (see Section 3.2) and our second tester (see Theorem 17 in Section 3.3) is slightly more refined, allowing for a more natural use later in the setting of streaming algorithms in Section 5.

We note that in this paper we focus on one specific model of access to the input graph, the *random-neighbor model.* It is possible to extend some of our analysis (of canonical testers) to some other graph access models, though (cf. the full version).

3.1 Random BFS and Bounded Discs

Property testing in query oracle model. Since we consider general graphs, without any bounds for vertex degrees, we have to carefully define the access provided to the input graph in the property testing framework. The access to the input graph is given by *queries* to an *oracle* representing the graph. There have been several oracles considered in the literature for general graphs, but our main focus is on the *random-neighbor model*, which we consider to be natural for graphs with unbounded degree, especially in the context of properties testable with a constant number of queries.

▶ **Definition 6** (Random-neighbor model). *In the* random-neighbor model*, an algorithm is given $n \in \mathbb{N}$ and access to an input graph $G = (V, E)$ by a query oracle, where $V = [n]$. The algorithm may ask queries based on the entire knowledge it has gained by the answers to previous queries. The* random neighbor query *specifies a vertex $v \in V$ and the oracle returns a vertex that is chosen i.u.r. (independently and uniformly at random) from the set of all neighbors of v.*

Notice that in the random-neighbor model, since $V = [n]$, the algorithm can also trivially select a vertex from V i.u.r. We believe that the random-neighbor model is the most natural model of computations in the property testing framework in the context of very fast algorithms (especially those of constant query complexity), and therefore our main focus is on that model. However, we want to point out that some of our results are sufficiently general to apply to a larger variety of the query oracle models, though we will not elaborate about it here (cf. the full version).

We describe the first canonical testers of all constant-query testers (in the random-neighbor model) for general graphs, both, for one-sided and two-sided errors. With this canonization, we can model all graph properties testable with a constant number of queries using *canonical testers*; see Theorems 10 and 17 for formal statements.

To formalize our canonical testers for all constant-query testers in the random-neighbor model, we will use the following two definitions of constrained random BFS-like graph exploration and of bounded discs.

We begin with the definition of a q-RBFS process, which starts at some vertex and explores its neighborhood in a BFS-like fashion, conditioned on a bound of the depth and the breadth of the exploration (see Definition 7 for formal definition and Algorithm 1 in Appendix B for the detailed implementation).

▶ **Definition 7** (q-random BFS). *Let $q > 0$ be an integer and G be a simple graph. For any vertex $v \in V(G)$, the q-*random BFS *(abbreviated as q-*RBFS*) explores a random subset of the q-neighborhood of v in G iteratively as follows. First, it initializes a queue $Q = \{v\}$*

and a graph $H = (\{v\}, \emptyset)$. Then, in every iteration, it pops a vertex u from Q and samples q random neighbors $s_{u,1}, \ldots, s_{u,q}$ of u. For every edge $e = \{u, s_{u,i}\}$, it adds $s_{u,i}$ and the directed edge $(u, s_{u,i})$ to H. Furthermore, if $s_{u,i}$ has distance less than q from v in H and $s_{u,i}$ has not been added to Q before, $s_{u,i}$ is appended to Q. When Q is empty, all edges in H are made undirected (without creating parallel edges) and H is returned.

Any output of q-RBFS algorithms can be described in a static form using the concept of bounded discs.

▶ **Definition 8** (q-bounded disc)**.** *For a given $q \in \mathbb{N}$, graph $G = (V, E)$, and vertex $v \in V$, a* q-bounded disc *of v* in *G is any subgraph H of G that is rooted at v and can be returned by* Random*BFS(G, v, q). In this case, vertex v is called a* root *of the q-bounded disc H and the maximum distance from v to any other vertex in H is called the* radius *of H.*

All q-bounded discs that are root-preserving isomorphic form an equivalence class.

▶ **Definition 9** (q-bounded disc type)**.** *Let H be a q-bounded disc. The equivalence class of H with respect to $\simeq$, i. e., the existence of a root-preserving isomorphism (see Definition 5), is called the q-bounded disc type of H.*

3.2 Canonical Testers: A General Version

Now we present the proof of our first main result. We show that any tester with query complexity $q = q(\varepsilon, n)$ in the random-neighbor model can be simulated by a *canonical tester* that samples $q' = O(q)$ vertices and rejects if and only if the union of the subgraphs induced by the q'-RBFS from the sampled vertices belongs to some family of forbidden graphs.

▶ **Theorem 10** (Canonical tester)**.** *Let $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ be a graph property that can be tested in the random-neighbor model with query complexity $q = q(\varepsilon, n)$ and error probability at most $\frac{1}{3}$. Then for every ε, there exists an infinite sequence $\mathcal{F} = (\mathcal{F}_n)_{n \in \mathbb{N}}$ such that for every $n \in \mathbb{N}$,*

- *$\mathcal{F}_n$ is a set of rooted graphs such that each graph $F \in \mathcal{F}_n$ is the union of q' many q'-bounded discs;*
- *the property Π_n on n-vertex graphs can be tested with error probability at most $\frac{1}{3}$ by the following canonical tester:*
  1. *sample q' vertices i.u.r. and mark them roots;*
  2. *for each sampled vertex v, perform a q'-RBFS starting at v;*
  3. *reject if and only if the explored subgraph is root-preserving isomorphic to some $F \in \mathcal{F}_n$,*

where $q' = cq$ for some constant $c > 1$. The query complexity of the canonical tester is $q^{O(q)}$. Furthermore, if $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ can be tested in the random-neighbor model with one-sided error, then the resulting canonical tester for Π has one-sided error too, i.e., the tester always accepts graphs satisfying Π.

3.3 Canonical Testers Revisited: Identifying Vertices in the Intersecting Discs

Theorem 10 provides us a canonical way of testing constant-query testable properties (in the random-neighbor model) by relating the tester to a set of forbidden subgraphs $\mathcal{F}_n$ for every $n \in \mathbb{N}$. However, as we mentioned in Section 1, it is hard to directly use Theorem 10 to design and analyze our streaming testers due to the intersections of q-RBFS. In order to tackle this difficulty, we decompose each forbidden subgraph $F \in \mathcal{F}_n$ into all possible sets of intersecting q-bounded discs whose union is F. In order to recover F from such a decomposition, we have to identify and monitor *vertices that are contained in more than one q-bounded disc of F.*

Identifying vertices with large reach probability. Now we prove that with constant probability the q-bounded discs found by q-RBFS will only intersect on a *small* set of vertices V_α and the discs will not intersect on any edge.

We begin with a useful definition on the probability of reaching a vertex from a q-RBFS.

▶ **Definition 11.** *For each vertex v, the* reach probability $r(v) := r_q(v)$ *of v is the probability that a q-RBFS starting at a uniformly randomly chosen vertex reaches v.*

In the following lemma, we give an upper bound on the size of the set of vertices with constant reach probability, which also implies that with constant probability, the number of vertices visited by at least two q-RBFS that the canonical tester performs is small. For any α, $0 \leq \alpha \leq 1$, we let $V_\alpha := \{v \in V : r(v) \geq \alpha\}$. For a fixed q, let $c_j := \sum_{i=0}^{j} q^i = \frac{q^{j+1}-1}{q-1}$.

▶ **Lemma 12.** *For any $0 < \alpha < 1$, it holds that $|V_\alpha| \leq \frac{c_q}{\alpha}$.*

We further show that with high probability, two q-RBFS starting from vertices chosen i.u.r. will not share an edge (i.e., will not visit the same edge).

▶ **Lemma 13.** *Let $0 < \alpha \leq 1$. Let $n \geq \frac{qc_q}{\alpha^2}$. Let u, v be two randomly chosen vertices. Let H_u and H_v denote the subgraphs visited by two q-RBFS starting at u and v, respectively. Then with probability at least $1 - qc_q \cdot 2\alpha$, no edge will be contained in both H_u and H_v.*

Colored q-bounded disc types. To identify vertices in V_α, we assign them unique colors for the analysis. We call a disc *r-colored* if in addition to uncolored vertices in the disc, some vertices in the disc may be colored with at most r colors, each color being used at most once. Two colored q-bounded disc types Δ_1 and Δ_2 (cf. Definition 9) are called to be isomorphic to each other, denoted by $\Delta_1 \simeq \Delta_2$, if there is a root-preserving isomorphism f from Δ_1 to Δ_2 that also preserves the colors, i.e., if and only if $u \in V(\Delta_1)$ is colored with color c, then $f(u) \in \Delta_2$ is colored with color c.

▶ **Definition 14.** *Let $q > 0$ be an integer. We let $\mathcal{H}_q := \{\Delta_1, \cdots, \Delta_N\}$ denote the set of all possible r-colored q-bounded disc types, where N is the total number of such types.*

For any given colored q-bounded disc type, we have the following definition on the probability of seeing such a disc type from a q-RBFS.

▶ **Definition 15** (Reach probability of colored q-bounded disc types). *Let $G = (V, E)$ be a graph with n vertices such that each vertex in V_α is assigned to a unique color. Let $\Delta \in \mathcal{H}_q$ be a colored q-bounded disc type. The* reach probability of Δ in G *is the probability that a q-RBFS from a random vertex in G reveals a graph that is (root- and color-preserving) isomorphic to Δ, that is* $\mathrm{Reach}_G(\Delta) := \Pr_{v \sim V, BFS}[\textit{RANDOMBFS}(G, v, q) \simeq \Delta]$.

For a given vertex v, the reach probability of Δ from v in G is the probability that a q-RBFS from v in G induces a graph that is (root- and color-preserving) isomorphic to Δ, that is $\mathrm{Reach}_G(v, \Delta) := \Pr_{BFS}[\textit{RANDOMBFS}(G, v, q) \simeq \Delta]$.

Recall from Definition 8 that a q-bounded disc of v in G is any subgraph H of G that is rooted at v and can be returned by $\textsc{RandomBFS}(G, v, q)$. In order to estimate the reach probability of a colored q-bounded disc type, we consider for each starting vertex v, the set of all possible colored q-bounded discs, denoted $\mathcal{C}_v$, that one can see from a q-RBFS from v.

▶ **Definition 16** (Reach probability of a colored q-bounded disc). *Let $G = (V, E)$ be a graph in which all vertices in V_α are uniquely colored. Let v be a vertex in G. A* colored q-bounded disc *of v is a q-bounded disc of v in G in which all vertices in V_α colored. We let $\mathcal{C}_v$ denote* *the set of all possible colored q-bounded discs of v.*[6] *For any fixed colored q-bounded disc $C \in \mathcal{C}_v$ of v, the reach probability of C from v is the probability that a q-RBFS from v sees exactly C, that is,* $\mathrm{Reach}_G(v, C) := \Pr_{BFS}[\textsc{Random}BFS(G, v, q) = C]$.

By our definition, the q-RBFS from a vertex v in the colored graph G (with vertices in V_α colored) will return exactly one colored q-bounded disc of v. For each colored q-bounded disc type Δ, we let $\mathcal{C}_v(\Delta)$ denote the subset of $\mathcal{C}_v$ which contains all colored q-bounded discs of v that are isomorphic to Δ. Therefore, we have the following observation: $\mathrm{Reach}_G(v, \Delta) = \sum_{D \in \mathcal{C}_v(\Delta)} \mathrm{Reach}_G(v, D)$.

Canonical testers with distinguished vertices in the intersecting discs. Now, we give a refined characterization of the family of forbidden subgraphs corresponding to any constant-query testable property in general graphs, which establishes the basis of our framework for transforming the canonical constant-query testers in the random-neighbor model to the random-order streaming model.

In our next theorem, we will consider partially vertex-colored graphs and q-bounded discs: we color each vertex in V_α with a unique color from a palette of size $|V_\alpha|$. Recall from Lemma 12 that $|V_\alpha| \le \frac{c_q}{\alpha}$. We obtain canonical testers of constant-query testable properties by *forbidden colored q-bounded discs* instead of *forbidden subgraphs* (that can be composed of more than a single q-bounded disc). See Figure 1 in Appendix A for an example.

▶ **Theorem 17.** *Let $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ be a graph property that is testable with query complexity $q = q(\varepsilon)$. Let $\alpha \le \frac{1}{24 q' c_{q'}}$, where q' is the number from Theorem 10 for a canonical tester with error probability $1/6$. There is an infinite sequence $\mathcal{F}' = (\mathcal{F}'_n)_{n \in \mathbb{N}}$ such that for any $\epsilon > 0$, $n \ge \frac{q' c_{q'}}{\alpha^2}$, the following properties hold:*

- *$\mathcal{F}'_n$ is a set of graphs, and for each graph $F \in \mathcal{F}'_n$, there exists at least one multiset S of q' many $c_{q'}/\alpha$-colored and rooted q'-bounded disc types such that 1) the disc types are pairwise edge-disjoint, and 2) the graph obtained by identifying all vertices of the same color in the bounded discs of S is isomorphic to F.*
- *For any n-vertex graph $G = (V, E)$ such that each vertex in V_α is colored uniquely, let $S_{q'}$ denote the set of q' subgraphs obtained by performing q'-RBFS starting at q' vertices sampled i.u.r. Then,*
  - *if $G \in \Pi_n$, with probability at least $\frac{2}{3}$, there is no $F \in \mathcal{F}'_n$ such that F is isomorphic to a graph from $S_{q'}$,*
  - *if G is ε-far from Π_n, with probability at least $\frac{2}{3}$, there exists $F \in \mathcal{F}'_n$ such that F is isomorphic to a graph from $\simeq S_{q'}$,*

  where the probability is taken over the randomness of $S_{q'}$.

Furthermore, if Π can be tested with one-sided error, then for $G \in \Pi_n$, with probability 1, there is no $F \in \mathcal{F}'_n$ such that $F \simeq S_{q'}$.

4 Estimating the Reach Probabilities in Random Order Streams

Given a canonical tester $\mathcal{T}$ for a property Π that is constant-query testable in the random-neighbor model, we transform it into a random-order streaming algorithm as follows. Recall from Theorem 10 that $\mathcal{T}$ explores the input graph by sampling vertices uniformly at random and running q-RBFS for each of these vertices. Only if the resulting subgraph contains an

[6] Note that the number $|\mathcal{C}_v|$ of colored q-bounded discs of v can be a polynomial of n.

instance of a forbidden subgraph from a family $\mathcal{F}$, it rejects. It seems natural to define a procedure like q-RBFS for random order streams, namely a procedure $\textsc{StreamCollect}(\mathcal{S}(G), v, q)$ (q-SC), and let the streaming algorithm reject only if the union of all q-SC contains an instance of a graph from $\mathcal{F}$. However, this raises a couple of issues.

It seems hard to analyze the union of the subgraphs obtained by q-SC and relate it to the union of subgraphs observed by q-RBFS because the interference between two q-SC is quite different from the interference of two q-RBFS. Therefore, we use Theorem 17, which roughly says that we can decompose each forbidden subgraph into colored q-bounded disc types. This leads to the following idea: First, we prove that for any colored q-bounded disc type Δ, if q-RBFS finds an instance of Δ in the input graph with probability p (where colors correspond to intersections of multiple RBFS), then q-SC finds an instance of Δ with probability cp for some suitable constant c. Then, we prove that if S is a sufficiently large set of vertices sampled uniformly at random, for each colored q-bounded disc type Δ, the fraction of q-bounded discs found by q-SCs started from S that are isomorphic to Δ is bounded from below by the probability that a q-RBFS from a random vertex sees a colored q-bounded disc that is isomorphic to Δ. Finally, in the next section, we conclude that if q-RBFS finds a forbidden subgraph $F \in \mathcal{F}$ with probability p, then the fraction of q-SC also finds this subgraph with probability cp (for some suitable constant c) because it will find the corresponding colored q-bounded discs that assemble F.

Collecting a q-Bounded Disc in a Stream. In our streaming algorithm, we need to collect a q-bounded disc from a vertex v. We do this in a natural and greedy way: We start with a graph $H = (U, F)$ with $U = \{v\}$ and $F = \emptyset$. Then whenever we see an edge (u, w) from the stream that is connected to our current graph H and adding (u, w) to H does not violate the q-bounded radius of H, and the degree of u or the degree of w in H is still less than q^{2q}, we add it to F (and possibly add one of its endpoints to U); otherwise, we simply ignore the edge. Note that the algorithm does not assign colors to the subgraphs it explores. The procedure $\textsc{StreamCollect}$ is formally defined in Algorithm 2 in Appendix B.

Relation of One q-SC and One q-RBFS. In the following, we show that for any vertex v, and any colored q-bounded disc C of v, the probability of collecting C from v by running $\textsc{StreamCollect}$ on a random order edge stream is at least a constant factor of the probability of reaching C from v by running a q-RBFS on G. The statements in this subsection hold for a single run of q-SC.

We emphasize that the coloring does not need to be explicitly given. It is sufficient if it can be applied when random access to the graph is given. In particular, we may assign each vertex in V_α a unique color. This enables us to identify the vertices where multiple q-RBFS may intersect, which is crucial to apply Theorem 17 later.

▶ **Lemma 18.** *Let G be a vertex-colored graph. There exists a constant $c_*(q)$ depending on q, such that for any colored q-bounded disc C of G, it holds that the probability (over $\mathcal{S}(G)$) that $\textsc{StreamCollect}(\mathcal{S}(G), v, q)$ contains C is at least $c_*(q) \cdot \mathrm{Reach}_G(v, C)$.*

The following lemma performs the step from q-bounded discs to q-bounded disc *types*.

▶ **Lemma 19.** *Let Δ be a fixed colored q-bounded disc type. Let $X_{v,\Delta}$ denote the indicator variable that $\textsc{StreamCollect}$ from v collects a subgraph that contains a colored q-bounded disc of v that is isomorphic to Δ. Let Y_v denote the indicator variable that $\textsc{RandomBFS}$ from v sees a colored q-bounded disc of v that is isomorphic to Δ. Then it holds that $\mathrm{E}_{\mathcal{S}(G)}[X_{v,\Delta}] \geq c_*(q) \cdot \mathrm{E}_{RBFS}[Y_v]$, where $c_*(q)$ is the constant from Lemma 18.*

Now we consider the probability of seeing a colored q-disc type Δ. Note that $\mathrm{E}_{\mathcal{S}(G)}[X_{v,\Delta}] = \Pr_{\mathcal{S}(G)}[\textsc{StreamCollect}(\mathcal{S}(G), v, q)$ contains a subgraph F with $F \simeq \Delta]$. Furthermore, it holds that $\mathrm{E}_{RBFS}[Y_v] = \mathrm{Reach}_G(v, \Delta)$. Thus, we have the following lemma.

▶ **Corollary 20.** *For any colored q-bounded disc type Δ, the probability (over $\mathcal{S}(G)$) that* $\textsc{StreamCollect}(\mathcal{S}(G),v,q)$ *contains a subgraph F with $F \simeq \Delta$ is at least* $c_*(q) \cdot \mathrm{Reach}_G(v, \Delta)$.

Relation of Multiple q-SCs and q-RBFS. In the above, we related a single run of q-RBFS and a single run of q-SC. In particular, Corollary 20 states that if a q-RBFS starting from v finds some colored q-bounded disc type Δ with probability p, q-SC finds the same type Δ with probability $\Omega(p)$. However, the forbidden subgraphs that the property tester aims to find may be composed of more than one q-bounded disc. Therefore, we need to prove that if multiple runs of q-RBFS find q-bounded disc types $\Delta_1, \dots, \Delta_k$ whose union contains an instance of a forbidden subgraph $F \in \mathcal{F}'_n$, then multiple runs of q-SC will find $\Delta_1, \dots, \Delta_k$ with probability $\Omega(p)$.

We now show our main technical lemma on estimating the reach probability of q-bounded disc types in random order streams. Again, the coloring of vertices in G is implicit and only used for the analysis.

▶ **Lemma 21.** *Let $G = (V, E)$ be a graph defined by a random order stream and let all vertices in V_α be colored. Let $q > 0$ be an integer and let $c'_q := \sum_{i=0}^{q+1} q^{2qi}$. Let $\delta > 0$, and let S denote a set of vertices that are chosen uniformly, where $s := |S| \geq \max\left\{\frac{1}{20\sqrt{\alpha q^{2q} \cdot c'_q}}, \frac{5000|\mathcal{H}_q|}{c_*(q)\delta^3}\right\}$, $\alpha := \frac{c_*(q)^4\delta^8}{10^9|\mathcal{H}_q|^2 q^{2q} c'_q}$. Let $\mathcal{J} := \{H_v : H_v = \textsc{StreamCollect}(\mathcal{S}(G), v, q), v \in S\}$ denote the set of colored q-bounded discs collected by* $\textsc{StreamCollect}$ *from vertices in S. For each type $\Delta \in \mathcal{H}_q$, let X_Δ denote the number of graphs H in $\mathcal{J}$ such that H contains a subgraph F with $F \simeq \Delta$.*

Then it holds that with probability at least $1 - \frac{1}{100}$, for each type $\Delta \in \mathcal{H}_q$, $q_\Delta := \frac{1}{c_*(q)} \cdot \frac{X_\Delta}{s} \geq$ $\mathrm{Reach}_G(\Delta) - \delta$ *, where $c_*(q)$ is a constant from Corollary 20.*

5 Testing Graph Properties in Random Order Streams

Now we transform constant-query property testers (with one-sided error) into constant-space streaming property testers, and prove Theorem 4. The main idea is to explore the streamed graph by $\textsc{StreamCollect}$ and look for the forbidden subgraphs in $\mathcal{F}_n$ that characterize Π (see Theorem 10). However, in the underlying analysis, we use the (reversible) decomposition of the forbidden subgraphs in $\mathcal{F}_n$ into $\mathcal{F}'_n$ (see Theorem 17) to prove the following: if $\mathcal{T}$ finds the colored q-bounded discs $\Delta_1, \dots, \Delta_k$ that compose a forbidden subgraph $F \in \mathcal{F}'_n$ with probability p, then the streaming tester will find at least as many copies of $\Delta_1, \dots, \Delta_k$ as $\mathcal{T}$ (see Lemma 21) and can stitch F from these copies. With these tools at hand, we can incorporate our analysis from previous sections to complete the proof of Theorem 4 (see Appendix C).

6 Conclusions

We gave the first canonical testers for all constant-query testers in the random-neighbor model for general graphs and show that one can emulate any constant-query tester with one-sided error in this query model in the random-order streaming model with constant space. Our transformation between constant-query testers and streaming algorithms with constant space provides a strong and formal evidence that property testing and streaming algorithms are very closely related. Our results also work for any restricted class of general graphs and other query models, e.g., random neighbor/edge model. It follows that many properties are constant-space testable (with one-sided error) in random order streams, including (s, t)-disconnectivity, being d-bounded degree, k-path-freeness of general graphs and bipartiteness and H-freeness of planar (or minor-free) graphs.

References

1 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It's all about regularity. *SIAM Journal on Computing*, 39(1):143–167, 2009. doi:10.1137/060667177.

2 Noga Alon, Tali Kaufman, Michael Krivelevich, and Dana Ron. Testing triangle-freeness in general graphs. *SIAM Journal on Discrete Mathematics*, 22(2):786–819, 2008. doi:10.1137/07067917X.

3 Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008. doi:10.1137/06064888X.

4 Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab Mirrokni, and Cliff Stein. Coresets meet EDCS: Algorithms for matching and vertex cover on massive graphs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1616–1635, 2019.

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

6 Marc Bury, Elena Grigorescu, Andrew McGregor, Morteza Monemizadeh, Chris Schwiegelshohn, Sofya Vorotnikova, and Samson Zhou. Structural results on matching estimation with applications to streaming. *Algorithmica*, 81(1):367–392, 2019.

7 Clément L. Canonne and Tom Gur. An adaptivity hierarchy theorem for property testing. *Computational Complexity*, 27(4):671–716, 2018. doi:10.1007/s00037-018-0168-4.

8 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 641–650, 2008.

9 Amit Chakrabarti, Prantar Ghosh, Andrew McGregor, and Sofya Vorotnikova. Vertex ordering problems in directed graph streams. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1786–1802, 2020.

10 Graham Cormode, Hossein Jowhari, Morteza Monemizadeh, and S. Muthukrishnan. The sparse awakens: Streaming algorithms for matching size estimation in sparse graphs. In *Proceedings of 25th Annual European Symposium on Algorithms (ESA)*, pages 29:1–29:15, 2017. doi:10.4230/LIPIcs.ESA.2017.29.

11 Artur Czumaj, Morteza Monemizadeh, Krzysztof Onak, and Christian Sohler. Planar graphs: Random walks and bipartiteness testing. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 423–432, 2011. doi:10.1109/FOCS.2011.69.

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

13 Artur Czumaj, Asaf Shapira, and Christian Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6):2499–2510, 2009.

14 Artur Czumaj and Christian Sohler. A characterization of graph properties testable for general planar graphs with one-sided error (It's all about forbidden subgraphs). In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1513–1536, 2019.

15 Hossein Esfandiari, Mohammadtaghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. *ACM Transactions on Algorithms*, 14(4):48, 2018.

16 Alireza Farhadi, Mohammad Taghi Hajiaghayi, Tung Mai, Anup Rao, and Ryan A. Rossi. Approximate maximum matching in random streams. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1773–1785, 2020.

17 Hendrik Fichtenberger, Pan Peng, and Christian Sohler. Every testable (infinite) property of bounded-degree graphs contains an infinite hyperfinite subproperty. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 714–726, 2019. doi:10.1137/1.9781611975482.45.

18 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

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

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

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

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

23 Monika Rauch Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. In *Proceedings of the DIMACS Workshop on External Memory Algorithms*, pages 107–118, 1998.

24 Zengfeng Huang and Pan Peng. Dynamic graph stream algorithms in $o(n)$ space. *Algorithmica*, 81(5):1965–1987, 2019.

25 Kazuo Iwama and Yuichi Yoshida. Parameterized testability. *ACM Transactions on Computation Theory*, 9(4):16, 2018.

26 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 734–751, 2014.

27 Michael Kapralov, Slobodan Mitrović, Ashkan Norouzi-Fard, and Jakab Tardos. Space efficient approximation to maximum matching size from uniform edge samples. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1753–1772, 2020.

28 Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1441–1483, 2004. doi:10.1137/S0097539703436424.

29 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Proceedings of the 15th International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 231–242, 2012.

30 Mitsuru Kusumoto and Yuichi Yoshida. Testing forest-isomorphism in the adjacency list model. In *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 763–774, 2014.

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

32 Andrew McGregor and Sofya Vorotnikova. A simple, space-efficient, streaming algorithm for matchings in low arboricity graphs. In *Proceedings of the 1st Symposium on Simplicity in Algorithms (SOSA)*, pages 14:1–14:4, 2018. doi:10.4230/OASIcs.SOSA.2018.14.

33 Morteza Monemizadeh, S. Muthukrishnan, Pan Peng, and Christian Sohler. Testable bounded degree graph properties are random order streamable. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 131:1–131:14, 2017.

34 Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

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

36 Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Structures & Algorithms*, 20(2):165–183, 2002. doi:10.1002/rsa.10013.

37 Pan Peng and Christian Sohler. Estimating graph parameters from random order streams. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2449–2466, 2018.

38 Sofya Raskhodnikova and Adam Smith. A note on adaptivity in testing properties of bounded degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(089), 2006.

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

40 Xiaoming Sun and David P Woodruff. Tight bounds for graph problems in insertion streams. In *Proceedings of the 18th International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 435–448, 2015.

41 Yuichi Yoshida and Yusuke Kobayashi. Testing the (s, t)-disconnectivity of graphs and digraphs. *Theoretical Computer Science*, 434:98–113, 2012.

A Missing Illustrations from Section 1

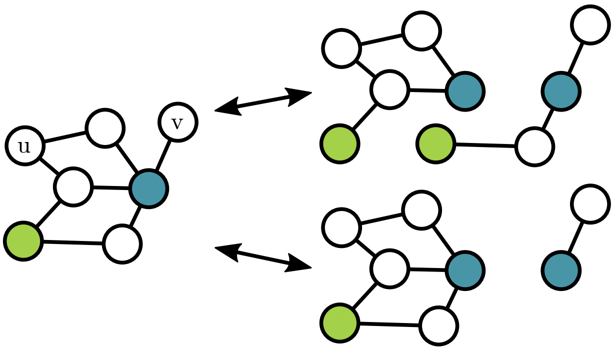

Figure 1 Consider the graph on the left, which can be decomposed into colored 3-bounded disc types (which are rooted at u and v in this example) in more than one way. However, it is always possible to recover the original graph by identifying vertices of the same color. Furthermore, every mapping is bijective because every color is assigned at most once per disc. If the colored vertices correspond to the vertices in V_α, every forbidden graph $F \in \mathcal{F}_n$ from Theorem 10 corresponds to a decomposition into edge-disjoint colored q-bounded discs $F' \in \mathcal{F}'_n$ in Theorem 17, which intersect only at colored vertices.

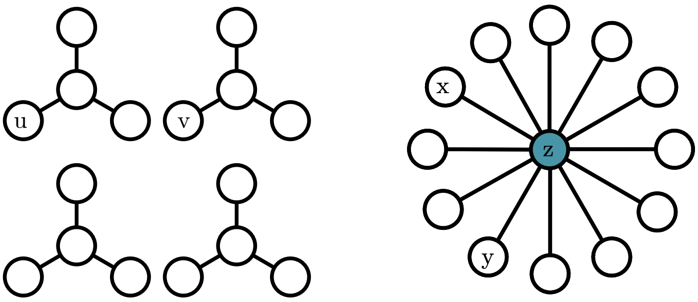

Figure 2 The above graph, which is composed of 3-stars and a $\omega(1)$-star with root z and which should be thought of as a subgraph of some larger graph, illustrates the need for colors in our analysis of the streaming property tester. Although the 2-bounded discs of u, v x and y are all 3-stars (with constant probability over the randomness of the neighbor queries), exploring u and v by q-RBFS does not result in finding a 6-star, while it is likely to find a 6-star by exploring x and y. Even if we prove that the probability that a q-SC finds uncolored 3-stars is lower bounded by some constant fraction of the probability that q-RBFS finds uncolored 3-stars, we still cannot rule out that q-SC might tend to find leaves of the small stars (like u and v) while q-RBFS tends to find leaves of the big star (like x and y). Observe that here, z is the only vertex that is likely contained in two different q-RBFS due to its high degree.

B Missing Pseudocodes from Section 3 and 4

The pseudocodes for the q-random BFS and for collecting a q-bounded disc from a vertex in stream are given below.

Algorithm 1 q-random BFS.

```
function RandomBFS($(G, v, q)$)
    $Q \leftarrow$ empty queue; enqueue$(Q, v)$
    $\forall w \in V : \ell[w] \leftarrow \infty$
    $\ell[v] \leftarrow 0$
    $H \leftarrow (\{v\}, \emptyset)$ with $v$ as root
    while $Q$ not empty do
        $u \leftarrow$ pop element from $Q$
        for $1 \le i \le q$ do
            $s_{u,i} \leftarrow$ query oracle for random neighbor of $u$
            add vertex $s_{u,i}$ and edge $(u, s_{u,i})$ to $H$
            if $(\ell[u] < q - 1) \wedge (\ell[s_{u,i}] = \infty)$ then
                $\ell[s_{u,i}] \leftarrow \ell[u] + 1$
                enqueue$(Q, s_{u,i})$
    return undirected $H$ without parallel edges
end function
```

Algorithm 2 Collecting a q-bounded disc from a vertex in stream.

```
function StreamCollect($(\mathcal{S}(G), v, q)$)
    $U \leftarrow \{v\}$
    $\forall u \in V : d_u \leftarrow 0, \ell_u \leftarrow \infty$
    $\ell_v \leftarrow 0; F \leftarrow \emptyset$
    $H = (U, F)$ with $v$ marked as root
    for $(u, w) \leftarrow$ next edge in the stream do
        if $(\{u, w\} \cap U \neq \emptyset)$ then
            if $(u \in U \Rightarrow (\ell_u < q \wedge d_u < q^{2q}) \vee (w \in U \Rightarrow (\ell_w < q \wedge d_w < q^{2q}))$ then
                $U \leftarrow U \cup \{u, w\}$
                $F \leftarrow F \cup (u, w)$
                $d_u \leftarrow d_u + 1; d_w \leftarrow d_w + 1$
                $\ell_u \leftarrow \min(\ell_u, \ell_w + 1); \ell_w \leftarrow \min(\ell_w, \ell_u + 1)$
    return $H$
end function
```

C Missing Proofs from Section 5

Proof of Theorem 4. We let $q_0 = q_0(\varepsilon)$ denote the query complexity of Π. Let $n = |V|$. We present our testing algorithm. Let $q = c \cdot q_0$ for some constant c from Theorem 10. Let $\alpha = \frac{c_*(q)^4\delta^8}{10^9|\mathcal{H}_q|^2 q^{2q} c'_q}$, where $c'_q = \sum_{i=0}^{q+1} q^{2qi}$, and $\delta = \frac{1}{200|\mathcal{H}_q|}$. If $n \le n_0 := \frac{qc_q}{\alpha^2}$, then we simply store the whole graph. If $n > n_0$, we proceed as follows. Let $\mathcal{F}_n$ be the set of forbidden subgraphs that characterize Π as stated in Theorem 10. We sample $s \ge \max\{\frac{1}{20\sqrt{\alpha q^{2q} \cdot c'_q}}, \frac{5000|\mathcal{H}_q|}{c_*(q)\delta^3}\}$ vertices $S \subseteq V$ and run StreamCollect$(\mathcal{S}(G), v, q)$ for each $v \in S$ to obtain a subgraph $H_v = (V_v, E_v)$ of G. If $H = \cup_{v \in S} H_v$ contains a forbidden subgraph $F \in \mathcal{F}_n$, the tester rejects, otherwise it accepts. See Algorithm 3 for details.

Algorithm 3 Testing graph property Π in random order stream.

```
function StreamTest($\mathcal{S}(G), n, \varepsilon, \mathcal{F}_n$)
    $S \leftarrow$ sample s vertices u.a.r. from $V$
    for all $v \in S$ do
        $H_v \leftarrow (V_v, E_v) = $ StreamCollect($\mathcal{S}(G), v, q$)
    $H \leftarrow (\cup_v V_v, \cup_v E_v)$
    if there exists $F \in \mathcal{F}_n$ such that $H$ contains a subgraph $F$ then
        Output Reject
    else
        Output Accept
end function
```

The space complexity of the algorithm is $s \cdot q_0^{O(q_0)} = O_{q_0}(1)$ words. For the correctness of the algorithm, we note that for any property Π that is constant-query testable with one-sided error, then with probability 1, we will not see any $F \in \mathcal{F}'_n$ if the graph G satisfies Π.

On the other hand, if G is ε-far from satisfying Π, then by Theorem 17, with probability at least $\frac{2}{3}$, the subgraph S_q spanned by the union of q-bounded discs rooted at q uniformly sampled vertices from G will span a subgraph that is isomorphic to some $F \in \mathcal{F}'_n$. Note that, in contrast to the algorithm above, the analysis uses the decomposition of forbidden subgraphs in $\mathcal{F}_n$ into colored q-discs given by Theorem 17. The key idea is to use the q-bounded discs that StreamCollect collects and the implicit colors (which are not observed by StreamCollect, but can be used in the analysis to identify vertices in V_α) to stitch forbidden subgraphs from $\mathcal{F}'_n$ that are discovered by RandomBFS. We prove that with sufficient probability, for each colored q-bounded disc Δ, StreamCollect finds at least as many copies of Δ as RandomBFS, and therefore, it can reproduce the same types of forbidden subgraphs from $\mathcal{F}'_n$.

By Markov's inequality and the union bound, the probability that at least one q-RBFS in the canonical tester for Π will return a colored q-bounded disc that is isomorphic to a disc Δ' such that $\text{Reach}_G(\Delta') < 2\delta = \frac{1}{100|\mathcal{H}_q|}$ is at most $\frac{1}{100}$. Let $\mathcal{D}$ be the set of all colored q-bounded discs Δ such that $\text{Reach}_G(\Delta) \geq 2\delta$.

By Lemma 21, with probability at least $1 - \frac{1}{100}$, for every $\Delta \in \mathcal{D}$, the number of graphs H_v obtained by StreamCollect that contain a subgraph isomorphic to Δ is at least $100|\mathcal{H}_q| \cdot \text{Reach}_G(\Delta) \geq 1$. By (implicitly) coloring all vertices in V_α, it follows from Theorem 17 that H contains a forbidden subgraph from $\mathcal{F}'_n$ with probability $1 - \frac{1}{100} - \frac{1}{100} > \frac{2}{3}$. ◀