Local Graph Exploration and Fast Property Testing*

Artur Czumaj

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

Abstract. We will present some recent results about testing graph properties in sparse graphs and will discuss graph exploration techniques which allow very efficient algorithms for testing graph properties.

1 Introduction

Property testing is a relaxation of classical decision problems, in which one wants to distinguish objects (for example, graphs, functions, or point sets) that have a given predetermined property (for example, being bipartite, monotone, or in convex position) from those that are *far* from this property. Property testing is motivated by the need to understand how to obtain information from massive structured or semi-structured data sets using small random samples. The notion of property testing was first explicitly formulated by Rubinfeld and Sudan [25] and it arises naturally in the context of program verification [4,25], learning theory, etc. Goldreich et al. [14] initiated the study of property testing for combinatorial objects, and in the recent years we have seen a number of property testing algorithms to test functions, probability distributions, graph and hypergraph properties, properties of languages, etc. (for references, see the surveys [8,13,23,24]).

One of the main directions in property testing is that of testing graph properties, as introduced by Goldreich, Goldwasser, and Ron [14,15]. The goal is for a given graph $G = (V, E)$ and a given property P (e.g., being bipartite), to decide if G satisfies property P or G is ε-far[1] from property P. Here, informally, we say *G is ε-far from property P* if it differs in an ε-fraction of its description from any graph having the property P. The tester is usually randomized and we allow it to err with probability at most $\frac{1}{3}$; if it always accepts any G that satisfies P then the tester has *one-sided error*; otherwise, it has *two-sided error*.

Property testing can be seen as approximation algorithms for decision problems. Since we only want to "approximately decide" problems, it is often possible to obtain algorithms that are much more efficient than their exact counterparts.

* Research supported by EPSRC award EP/G064679/1 and by the Centre for Discrete Mathematics and its Applications (DIMAP), EPSRC award EP/D063191/1.

[1] Normally one considers ε as being a small constant, independent of the input.

M. de Berg and U. Meyer (Eds.): ESA 2010, Part I, LNCS 6346, pp. 410–414, 2010.

As it has been demonstrated in numerous papers in the last decade, many property testers have indeed running times that are sublinear in the input size; for some problems it is even possible to obtain property testers whose complexity is *independent of the input size.* This, in turn, resulted in the recent development of sublinear-time approximation algorithms for many classical combinatorial problems, including dense max-cut, clustering problems, and estimating the cost of the minimum spanning tree and maximum matching (see, e.g., [5,6,9,10,12,14,22] and [8]).

1.1 Testing Properties of Dense Graphs

Since the notion of being ε-far depends on the input representation, we consider here two most popular models for analyzing graph properties. We begin with the *adjacency matrix model* [14], where the input graph $G = (V, E)$ on n vertices is represented by its adjacency matrix of size n^2; then G *is* ε*-far from property* P if one has to modify at least εn^2 entries in the adjacency matrix to obtain a graph satisfying property P. (Since we consider here graphs obtained by changing $\mathcal{O}(\varepsilon n^2)$ edges of the original graph G, it is easy to see that this definition is suitable mostly for dense graphs.) For such input representation, the complexity of a testing algorithm is measured by the number of queries to the adjacency matrix of G.

This model has been extensively studied in the last decade and it is now quite well understood [13]. After a series of papers, we know that in the adjacency matrix model, testability of a property in constant time (independent on the input size, but possibly depending on ε) is closely related to Szemerédi partitions of the graph. In fact, it has been shown that a graph property is testable in time independent of the size of G if and only if it can be reduced to testing finitely many Szemerédi partitions [1]. Moreover, Alon and Shapira [2] show that any natural graph property is testable in time independent of n with one-sided error if and only if it is either hereditary or it is close (in some well-defined sense) to a hereditary property.

1.2 Testing Properties of Sparse Graphs and Graph Exploration

While property testing in the adjacency matrix model for dense graphs is now relatively well understood, much less is known about testing properties of sparse (or arbitrary) graphs in the *adjacency list model.*

Properties of sparse graphs have been traditionally studied in the model of *bounded-degree* graphs introduced by Goldreich and Ron [15]. In this model, the input graph $G = (V, E)$ is represented by its adjacency list (or, equivalently, by its incidence list) and the vertex degrees are bounded by a constant d independent of the number of vertices of G. A testing algorithm has a constant-time access to any entry in the adjacency list by making a query to the i^{th} neighbor of a given vertex, and the number of accesses to the adjacency list is the (query) *complexity of the tester.* A property testing algorithm is an algorithm that for a given graph G determines if it satisfies a predetermined property P or it is ε-far from property P; a graph G is ε-far from property P if one has to modify more than εdn edges in G to obtain a graph having property P.

We will now briefly discuss central results for testing properties of sparse graphs and basic techniques used to efficiently test properties of sparse graphs.

Goldreich and Ron [15,16,17] were the first to study properties of sparse graphs. They show that although some basic properties (eg., being connected, k-connected, or Eulerian) can be tested in time independent of n, a number of fundamental properties require a superconstant testing time. For example, it has been shown that testing if a graph G is bipartite requires $\Omega(\sqrt{n})$ time [15]. (One needs $\Omega(\sqrt{n})$ time to distinguish between random graphs from the following two classes: *(i)* sum of a Hamiltonian cycle H and a perfect matching, and *(ii)* sum of a Hamiltonian cycle H and a perfect matching M such that each edge from M creates an even-length cycle when added to H. Since the latter class consists only of bipartite graphs and a random graph from the former class is with high probability ε-far from bipartite, the lower bound follows.) Similar bounds are known for testing if a graph is a good expander, is k-colorable, etc.

Comparing to the results for the adjacency matrix model, the adjacency list model has much more algorithmic flavor. In particular, to test if a given graph has a predetermined property one typically requires to do much more than just a simple sampling of vertices; many testing algorithms require some graph exploration algorithms to collect information about local and global properties of graphs. Indeed, the main three techniques are random sampling, local search (exploring the neighborhood of a vertex), and random walks.

The aforementioned lower bound for testing bipartitness was complemented by an upper bound of $\widetilde{\mathcal{O}}(\sqrt{n}/\varepsilon^{\mathcal{O}(1)})$ time for testing bipartitness in [16]. The algorithm by Goldreich and Ron is very representative for the area:

```
Testing Bipartitness:

– Pick a random sample $S$ of $\mathcal{O}(1/\varepsilon)$ vertices
– For each $v \in S$:
    • perform $\mathrm{poly}(\log n/\varepsilon)\sqrt{n}$ random walks from $v$, each of
      length $\mathrm{poly}(\log n/\varepsilon)$
    • If the graph induced by all edges visited is not bipartite
      then reject
– If the algorithm did not reject yet then accept
```

The analysis of the algorithm is very elaborate and it establishes an interesting connection with the analysis of the convergence times of Markov chains. Similar techniques of exploring the input graph by many independent random walks have been used in several recent works, for example, in $\widetilde{\mathcal{O}}(\sqrt{n}/\varepsilon^{\mathcal{O}(1)})$-time algorithms for testing if a graph is a good expander [11,19,21] (see also [17]).

A new approach has been proposed recently by Czumaj, Sohler, and Shapira [7], who consider testing graph properties not for all graphs, but rather for some specific classes of graphs. For example, they show that if the underlying graph is planar, then any hereditary graph property (e.g., bipartiteness, k-colorability, or perfectness) is testable in time independent of the input size. This approach can be generalized to any class of graphs that can be partitioned into constant size components by removing εn edges of the graph, for any $\varepsilon > 0$; we call such graphs *hyperfinite*. Benjamini et al. [3] extended this result and show that every minor-closed graph property is testable in time independent of the input size in general bounded-degree graphs (with two-sided error); this is shown by first proving that hyperfiniteness is testable for general bounded-degree graphs, and then by observing that every minor-closed graph property is hyperfinite. In particular, testing if a graph is planar can be done in time independent on n with two-sided error. This result (the complexity) has been improved by Hassidim et al. [18]; in particular, testing if a bounded degree graph is planar can be done with $2^{\mathrm{poly}(1/\varepsilon)}$ queries. Hassidim et al. [18] show also how to apply similar techniques to not only test graph properties, but also to approximate the distance to almost any hereditary property in any bounded degree hereditary families of graphs.

Future directions and challenges. Despite seeing a lot of progress in the last several years, we still have only a partial picture of the complexity of testing graph properties in the model of sparse graphs. For example, we know that in the two-sided error model, testing planarity can be done in time $2^{\mathrm{poly}(1/\varepsilon)}$; can we do it in time polynomial in $1/\varepsilon$? Or, what is the complexity of testing planarity in the one-sided error model, where it is conjectured that the complexity is $\Theta(\sqrt{n}/\varepsilon^{\mathcal{O}(1)})$. Similar question can be asked for testing other minor-closed properties. In fact, a more ambitious challenge would be to provide a characterization of properties of bounded degree graphs represented by adjacency lists that can be tested in constant time, or in $\Theta(\sqrt{n}/\varepsilon^{\mathcal{O}(1)})$ time. In this flavor, it has been shown recently that testing (with one-sided error) the property of having an H-minor can be done in time independent of n if and only if H is cycle-free.

The results for sparse graphs mentioned above are dealing solely with bounded-degree graphs. While the model of bounded-degree graphs is elegant and very natural, it is also desirable to consider the model of arbitrary graphs represented by adjacency list (see, e.g., [20]). How quickly can we test basic graph properties in this model? For example, it is easy to see that testing planarity require $\Omega(\sqrt{n})$; can we design an algorithm that could match this bound?

References

1. Alon, N., Fischer, E., Newman, I., Shapira, A.: A combinatorial characterization of the testable graph properties: it's all about regularity. SIAM Journal on Computing 39(1), 143–167 (2009)
2. Alon, N., Shapira, A.: A characterization of the (natural) graph properties testable with one-sided error. SIAM Journal on Computing 37(6), 1703–1727 (2008)
3. Benjamini, I., Schramm, O., Shapira, A.: Every minor-closed property of sparse graphs is testable. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC), pp. 393–402 (2008)
4. Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. Journal of Computer and System Sciences 47(3), 549–595 (1993)

5. Chazelle, B., Rubinfeld, R., Trevisan, L.: Approximating the minimum spanning tree weight in sublinear time. SIAM Journal on Computing 34(6), 1370–1379 (2005)
6. Czumaj, A., Ergün, F., Fortnow, L., Magen, A., Newman, I., Rubinfeld, R., Sohler, C.: Approximating the weight of the Euclidean minimum spanning tree in sublinear time. SIAM Journal on Computing 35(1), 91–109 (2005)
7. Czumaj, A., Shapira, A., Sohler, C.: Testing hereditary properties of nonexpanding bounded-degree graphs. SIAM Journal on Computing 38, 2499–2510 (2009)
8. Czumaj, A., Sohler, C.: Sublinear-time algorithms. Bulletin of the EATCS 89, 23–47 (2006)
9. Czumaj, A., Sohler, C.: Sublinear-time approximation algorithms for clustering via random sampling. Random Structures and Algorithms 30(1-2), 226–256 (2007)
10. Czumaj, A., Sohler, C.: Estimating the weight of metric minimum spanning trees in sublinear-time. SIAM Journal on Computing 39(3), 904–922 (2009)
11. Czumaj, A., Sohler, C.: Testing expansion in bounded-degree graphs. In: Proceedings of the 48th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 570–578 (2007)
12. Frieze, A., Kannan, R.: The regularity lemma and approximation schemes for dense problems. In: Proceedings of the 37th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 12–20 (1996)
13. Goldreich, O.: Introduction to testing graph properties. Electronic Colloquium on Computational Complexity (ECCC), Report No. 082 (2010)
14. Goldreich, O., Goldwasser, S., Ron, D.: Property testing and its connection to learning and approximation. Journal of the ACM 45(4), 653–750 (1998)
15. Goldreich, O., Ron, D.: Property testing in bounded degree graphs. Algorithmica 32(2), 302–343 (2002)
16. Goldreich, O., Ron, D.: A sublinear bipartiteness tester for bounded degree graphs. Combinatorica 19(3), 335–373 (1999)
17. Goldreich, O., Ron, D.: On testing expansion in bounded-degree graphs. Electronic Colloquium on Computational Complexity (ECCC), Report No. 7 (2000)
18. Hassidim, A., Kelner, J.A., Nguyen, H.N., Onak, K.: Local graph partitions for approximation and testing. In: Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 22–31 (2009)
19. Kale, S., Seshadhri, C.: An expansion tester for bounded degree graphs. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 527–538. Springer, Heidelberg (2008)
20. Kaufman, T., Krivelevich, M., Ron, D.: Tight bounds for testing bipartiteness in general graphs. SIAM Journal on Computing 33(6), 1441–1483 (2004)
21. Nachmias, A., Shapira, A.: Testing the expansion of a graph. Information and Computation 208(4), 309–314 (2010)
22. Nguyen, H.N., Onak, K.: Constant-time approximation algorithms via local improvements. In: Proceedings of the 49th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 327–336 (2008)
23. Ron, D.: Property testing. In: Handobook of Randomized Algorithms, vol. II, pp. 597–649 (2001)
24. Rubinfeld, R.: Sublinear time algorithms. In: Proceedings of the International Congress of Mathematicians, ICM (2006)
25. Rubinfeld, R., Sudan, M.: Robust characterization of polynomials with applications to program testing. SIAM Journal on Computing 25(2), 252–271 (1996)