Algorithmic and Analysis Techniques in Property Testing

Algorithmic and Analysis Techniques in Property Testing

Dana Ron

School of EE
Tel-Aviv University
Ramat Aviv, 10027
Israel
danar@eng.tau.ac.il

the essence of knowledge

Boston – Delft

Foundations and Trends® in Theoretical Computer Science

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is D. Ron, Algorithmic and Analysis Techniques in Property Testing, Foundation and Trends® in Theoretical Computer Science, vol 5, no 2, pp 73–205, 2009

ISBN: 978-1-60198-318-3

Foundations and Trends® in Theoretical Computer Science
Volume 5 Issue 2, 2009

Editorial Board

Editorial Scope

Foundations and Trends® in Theoretical Computer Science will publish survey and tutorial articles in the following topics:

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations Research
- Parallel algorithms
- Quantum Computation
- Randomness in Computation

Information for Librarians
Foundations and Trends® in Theoretical Computer Science, 2009, Volume 5, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.

Foundations and Trends® in
Theoretical Computer Science
Vol. 5, No. 2 (2009) 73–205

DOI: 10.1561/0400000029

now
the essence of knowledge

Algorithmic and Analysis Techniques in Property Testing

Dana Ron*

School of EE, Tel-Aviv University, Ramat Aviv, 10027, Israel,
danar@eng.tau.ac.il

Abstract

Property testing algorithms are "ultra"-efficient algorithms that decide whether a given object (e.g., a graph) has a certain property (e.g., bipartiteness), or is significantly different from any object that has the property. To this end property testing algorithms are given the ability to perform (local) queries to the input, though the decision they need to make usually concerns properties with a global nature. In the last two decades, property testing algorithms have been designed for many types of objects and properties, amongst them, graph properties, algebraic properties, geometric properties, and more.

In this monograph we survey results in property testing, where our emphasis is on common analysis and algorithmic techniques. Among the techniques surveyed are the following:

- The *self-correcting* approach, which was mainly applied in the study of property testing of algebraic properties;

* This work was supported by the Israel Science Foundation (grant number 246/08).

- The *enforce-and-test* approach, which was applied quite extensively in the analysis of algorithms for testing graph properties (in the dense-graphs model), as well as in other contexts;
- Szemerédi's *Regularity Lemma*, which plays a very important role in the analysis of algorithms for testing graph properties (in the dense-graphs model);
- The approach of *Testing by implicit learning*, which implies efficient testability of membership in many functions classes; and
- Algorithmic techniques for testing properties of sparse graphs, which include *local search* and *random walks*.

Contents

1 Introduction **1**

1.1 Settings in Which Property Testing is Beneficial 2
1.2 A Brief Overview 4
1.3 Property Testing and "Classical" Approximation 6
1.4 Property Testing and Learning 7
1.5 Organization of this Survey 8
1.6 Related Surveys 9

2 Preliminaries **11**

2.1 Basic Definitions and Notations 11
2.2 Testing Graph Properties 13

3 The Self-correcting Approach **17**

3.1 Linearity 18
3.2 Low-Degree Polynomials 24
3.3 Implications of Self-correction 27

4 The Enforce-and-Test Approach **37**

4.1 Testing Whether a Graph is a Biclique 37
4.2 Testing Bipartiteness in the Dense-Graphs Model 39
4.3 Other Applications of the Enforce-and-Test Approach 47

5 Testing by Implicit Learning **49**

5.1 A Building Block: Testing Juntas 50
5.2 The Algorithm for Testing by Implicit Learning 57

6 The Regularity Lemma **63**

6.1 Background 63
6.2 Statement of the Lemma 64
6.3 Testing Triangle-Freeness 65

7 Local-Search Algorithms **69**

7.1 Connectivity 70
7.2 k-Edge Connectivity 74
7.3 k-Vertex Connectivity. 78
7.4 Minor-Closed Properties 78
7.5 Other Local-Search Algorithms 80

8 Random Walks Algorithms **83**

8.1 Testing Bipartiteness in Bounded-Degree Graphs 83
8.2 Testing Expansion 88

9 Lower Bounds **91**

9.1 A Lower Bound for Testing Triangle-Freeness 91
9.2 A Lower Bound for Testing Bipartiteness of Constant Degree Graphs 96

10 Other Results **101**

10.1 Testing Monotonicity 101
10.2 Testing in the General-Graphs Model 109
10.3 Testing Membership in Regular Languages and Other Languages 111

11 Extensions, Generalizations, and Related Problems **115**

11.1 Distribution-Free Testing 115
11.2 Testing in the Orientation Model 116
11.3 Tolerant Testing and Distance Approximation 118
11.4 Testing and Estimating Properties of Distributions 119

Acknowledgments **129**

References **131**

1

Introduction

Property testing algorithms are algorithms that perform a certain type of *approximate decision*. Namely, standard (exact) decision algorithms are required to determine whether a given input is a YES instance (has a particular property) or is a NO instance (does not have the property). In contrast, property testing algorithms are required to determine (with high success probability) whether the input has the property (in which case the algorithm should *accept*) or is *far* from having the property (in which case the algorithm should *reject*). In saying that the input is *far* from having the property we mean that the input should be modified in a non-negligible manner so that it obtains the property.

To be precise, the algorithm is given a *distance* parameter, denoted ϵ, and should reject inputs that are ϵ-far from having the property (according to a prespecified distance measure). If the input neither has the property nor is far from having the property, then the algorithm can either accept or reject. In other words, if the algorithm accepts, then we know (with high confidence) that the input is close to having the property, and if it rejects, then we know (with high confidence) that the input does not have the property.

Since a property testing algorithm should perform only an approximate decision and not an exact one, we may expect it to be (*much*) more efficient than any exact decision algorithm for the same property. In particular, as opposed to exact decision algorithms, which are considered efficient if they run in time that is polynomial in the size of the input (and the best we can hope for is linear-time algorithms), property testing algorithms may run in time that is *sublinear* in the size of the input (and hence we view them as being "ultra"-efficient). In such a case they cannot even read the entire input. Instead, they are given *query access* to the input, where the form of the queries depends on the type of input considered.

Since property testing algorithms access only a small part of the input, they are naturally allowed to be randomized and to have a small probability of error (failure). In some cases they have a non-zero error probability only on inputs that are far from having the property (and *never* reject inputs that have the property). In such a case, when they reject an input, they always provide (small) *evidence* that the input does not have the property.

By the foregoing discussion, when studying a specific property testing problem, one should define a distance measure over inputs (which determines what inputs should be rejected), and one should define the queries that the algorithm is allowed. For example, when dealing with functions and their properties (e.g., linearity), the distance measure is usually defined to be the Hamming distance normalized by the size of the domain, and queries are simply queries for values of the function at selected elements of the domain. In other cases, such as graph properties, there are several different natural models for testing (see Section 2.2 for details).

1.1 Settings in Which Property Testing is Beneficial

In addition to the intellectual interest in relating global properties to local patterns, property testing algorithms are beneficial in numerous situations. A number of such settings are discussed next.

1. *Applications that deal with huge inputs.* This is the case when dealing with very large databases in applications related to

computational biology, astronomy, study of the Internet, and more. In such cases, reading the entire input is simply infeasible. Hence, some form of approximate decision, based on accessing only a small part of the input, is crucial.

2. *Applications in which the inputs are not huge, but the property in question is $\mathcal{NP}$-hard.* Here too some form of approximation is necessary, and property testing algorithms provide one such form. In fact, while "classical" approximation algorithms are required to run in time polynomial in the size of the input, here we require even more of the algorithm: It should provide an approximately good answer, but is allowed only sublinear time. For example, there is a property testing algorithm that can be used to obtain a $(1 \pm \epsilon)$-factor approximation of the size of the maximum cut in a dense graph, whose running time depends only on ϵ, and does not depend at all on the size of the graph. (In Section 1.3 we further discuss the relation between the notion of approximation provided by property testing and more "classical" notions.)
3. *Applications in which the inputs are not huge and the corresponding decision problem has a polynomial-time algorithm, but we are interested in ultra-efficient algorithms, and do not mind sacrificing some accuracy.* For example, we may not mind accepting a graph that is not *perfectly* bipartite, but is *close* to being bipartite (that is, it has a two-way partition with relatively few "violating edges" within the two parts).
4. *Scenarios similar to the one described in the previous item except that the final decision must be exact* (though a small probability of failure is allowed). In such a case we can first run the testing algorithm, and only if it accepts do we run the exact decision procedure. Thus, we save time whenever the input is far from having the property, and this is useful when typical (but not all) inputs are far from having the property. A related scenario, discussed in Section 1.4, is the application of property testing as a preliminary step to learning.

Thus, employing a property testing algorithm yields a certain loss in terms of accuracy, but our gain, in terms of efficiency, is in many cases dramatic. Furthermore, in many cases the loss in accuracy is inevitable either because the input is huge or the problem is hard.

1.2 A Brief Overview

Property testing first appeared (implicitly) in the work of Blum et al. [35], who designed the well-known *Linearity testing algorithm*. It was first explicitly defined in the work of Rubinfeld and Sudan [123], who considered testing whether a function is a low-degree polynomial. The focus of these works was on testing algebraic properties of functions, and they, together with other works, had an important role in the design of *Probabilistically Checkable Proofs* ($\mathcal{PCP}$) systems (cf. [19, 20, 21, 22, 57, 66, 67, 123]).

The study of property testing in a more general context was initiated by Goldreich et al. [72]. They gave several general results, among them results concerning the relation between testing and learning, and then focused on testing properties of graphs (in what we refer to as the *dense-graphs* model). Following this work, property testing has been applied to many types of inputs and properties.[1] In particular, the study of algebraic properties of functions continued to play an important role, partly because of the relation to the area of *error correcting codes* (for a short explanation concerning this relation, see the beginning of Section 3). The study of graph properties was significantly extended since the work of Goldriech et al. [72]. This includes a large number of works in the dense-graphs model, as well as the introduction of other models (more suitable for graphs that are sparse or that are neither dense nor sparse), and the design of algorithms that work within these models. There has also been progress in the last few years on the design of testing algorithms for properties of functions that can be viewed as *logical* rather than algebraic (such as functions that have a small DNF representation). The study of such properties is of interest from the point of view of learning theory (see Section 1.4). Other families of properties to

[1] In what follows in this subsection we do not give references to relevant works. These references can be found in the body of this monograph when each specific result is mentioned.

which the framework of property testing has been applied include Geometric properties and "clusterability" of ensembles of points, properties defined by restricted languages (e.g., regular languages), properties of distributions, and more.

In some cases the algorithms designed are extremely efficient: The number of operations they perform *does not depend* at all on the size of the input, but only on the distance parameter ϵ. In other cases the dependence is some sublinear function of the size of the input (e.g., polylog(n) or $\sqrt{n}$, for inputs of size n), where in many of the latter cases there are matching (or almost matching) lower bounds that justify this dependence on the size of the input.

While each algorithm has features that are specific to the property it tests, there are several common algorithmic and analysis techniques. Perhaps, the two better-known analysis techniques are the *self-correcting* approach, which is applied in the analysis of many testing algorithms of algebraic properties, and Szemerédi's Regularity Lemma [124], which is central to the analysis of testing graph properties in the dense-graphs model. Other techniques include the *enforce-and-test* approach (that is also applied in the analysis of testing algorithms in the dense-graphs model, as well as in testing certain metric properties and clustering properties), and the approach of *testing by implicit learning* whose application gives a variety of results (among them testing of small DNF formula). Indeed, as the title of this monograph suggests, we organize the results presented according to such common techniques.

In addition to the extension of the scope of property testing, there have been several extensions and generalizations of the basic notion of property testing. One extension (which was already introduced in [72] but for which positive results appeared several years later) is allowing the underlying distribution (with respect to which the distance measure is defined) to be different from the uniform distribution (and in particular to be unknown — this is referred to as *distribution-free* testing). Another natural extension is to *tolerant testing*. In tolerant testing the algorithm is given two distance parameters: ϵ_1 and ϵ_2, and it must distinguish between the case that the object is ϵ_1-close to having the property (rather than perfectly having the property as in the original definition of property testing) and the case that the object is ϵ_2-far from

having the property. A related notion is that of *distance approximation* where the task is to obtain an estimate of the distance to having the property.

1.3 Property Testing and "Classical" Approximation

Consider for example the problem of deciding whether a given graph $G = (V, E)$ has a clique of size at least k, for $k = \rho n$ where ρ is a fixed constant and $n = |V|$. The "classical" notion of an approximation algorithm for this problem requires the algorithm to distinguish between the case that the max-clique in the graph has size at least ρn and, say, the case in which the max-clique has size at most $\rho n/2$.

On the other hand, when we talk of testing the "ρ-Clique" property, the task is to distinguish between the case that the graph has a clique of size ρn and the case in which it is ϵ-far from the any n-vertex graph that has a clique of size ρn. Since this property is relevant only to dense graphs (where $|E| = \Theta(n^2)$), our notion of ϵ-far in this context is that more than ϵn^2 edges should be added to the graph so that it has a clique of size ρn. This is equivalent to the *dual approximation* task (cf., [89, 90]) of distinguishing between the case that an n-vertex graph has a clique of size ρn and the case that in any subset of ρn vertices, the number of missing edges (between pairs of vertices in the subset) is more than ϵn^2.

The above two tasks are vastly different: Whereas the former task is $\mathcal{NP}$-hard, for $\rho < 1/4$ [30, 88], the latter task can be solved in $\exp(O(1/\epsilon^2))$-time, for any $\rho, \epsilon > 0$ [72]. We believe that there is no absolute sense in which one of these approximation tasks is better than the other: Each of these tasks is relevant in some applications and irrelevant in others. We also mention that in some cases the two notions coincide. For example, consider the problem of deciding whether a graph has a cut of size at least k for $k = \rho n^2$ (where ρ is a fixed constant). Then a testing algorithm for this problem will distinguish (with high probability) between the case that the max-cut in the graph is of size at least ρn^2 and the case in which the max-cut is of size less than $(\rho - \epsilon)n^2$ (which for $\epsilon = \gamma\rho$ gives a "classical" $(1 - \gamma)$-factor approximation to the size of the max-cut).

Finally, we note that while property testing algorithms are decision algorithms, in many cases they can be transformed into optimization algorithms that actually construct approximate solutions. To illustrate this, consider the two aforementioned properties, which we refer to as ρ-Clique and ρ-Cut. For the first property, suppose the graph has a clique of size at least ρn. Then, building on the testing algorithm, it is possible to obtain (with high probability (w.h.p.)), in time that grows only linearly in n, a subset of ρn vertices that is close to being a clique. (That is, the number of missing edges between pairs of vertices in the subset is at most ϵn^2.) Similarly, for the second property, if the graph has a cut of size at least ρn^2, then it is possible to obtain (w.h.p.), in time linear in n, a cut of size at least $(\rho - \epsilon)n^2$. In both cases the dependence on $1/\epsilon$ in the running time is exponential (whereas a polynomial dependence cannot be obtained unless $\mathcal{P} = \mathcal{NP}$).

For these problems and other partition problems (e.g., k-colorability), the testing algorithm (when it accepts the input) actually defines an *implicit* partition. That is, after the execution of the testing algorithm, it is possible to determine for each vertex (separately) to which part it belongs in the approximately good partition, in time poly$(1/\epsilon)$.

1.4 Property Testing and Learning

Following standard frameworks of learning theory, and in particular the PAC learning model of Valiant [125] and its variants, when we say *learning* we mean outputting a good estimate of a function to which we have query access (or from which we can obtain random labeled examples). Thus, another view of property testing is as a relaxation of learning (with queries and under the uniform distribution).[2] Namely, instead of asking that the algorithm output a good estimate of the (target) function (which is possibly assumed to belong to a particular class of functions $\mathcal{F}$), we only require that the algorithm decide whether the function belongs to $\mathcal{F}$ or is far from any function in $\mathcal{F}$. Given

[2] Testing under non-uniform distributions and testing with random examples (only) have been considered (and we discuss the former in this monograph), but most of the work in property testing deals with testing under the uniform distributions and with queries.

this view, a natural motivation for property testing is to serve as a preliminary step before learning: We can first run the testing algorithm in order to decide whether to use a particular class of functions as our hypothesis class.

In this context too we are interested in testing algorithms that are more efficient than the corresponding learning algorithms. As observed in [72], property testing is no harder than *proper* learning (where the learning algorithm is required to output a hypothesis from the same class of functions as the target function). Namely, if we have a proper learning algorithm for a class of functions $\mathcal{F}$, then we can use it as a subroutine to test the property of membership in $\mathcal{F}$.

We also note that property testing is related to *hypothesis testing* (see e.g., [101, Chap. 8]). For a short discussion of this relation, see the introduction of [121].

1.5 Organization of this Survey

In this monograph we have chosen to present results in property testing with an emphasis on analysis techniques and algorithmic techniques. Specifically:

- In Section 3 we discuss results whose analysis follows the *Self-correcting* approach (e.g., testing linearity), and mention several implications of this approach.
- In Section 4 we discuss results whose analysis follows the *enforce-and-test* approach (e.g., testing bipartiteness in the dense-graphs model). In many cases this approach implies that the testing algorithm can be transformed into an efficient approximate optimization algorithm (as discussed in Section 1.3).
- The approach of *Testing by Implicit Learning*, whose application leads to efficient testing of many function classes (e.g., DNF formula with a bounded number of terms), is described in Section 5.
- The *Regularity Lemma* of Szemerédi [124], which is a very important tool in the analysis of testing algorithms in the

dense-graphs model, is presented in Section 6, together with its application to testing triangle-freeness (in this model).
- In Section 7 we discuss algorithms for testing properties of sparse graphs that are based on *local search*.
- The use of *random walks* by testing algorithms for properties of sparse graphs is considered in Section 8.
- In Section 9 we present two examples of lower bound proofs for property testing algorithms, so as to give a flavor of the type of arguments used in such proofs.
- A small selection of other families of results, which did not fit naturally in the previous sections (e.g., testing monotonicity of functions), is discussed in Section 10.
- We conclude the monograph in Section 11 with a discussion of several extensions and generalizations of property testing (e.g., tolerant testing).

1.6 Related Surveys

There are several surveys on property testing ([58, 69, 120], and the more recent [121]), which have certain overlaps with the current survey. In particular, the recent survey [121] of the current author presents property testing from a learning theory perspective. Thus, the emphasis in that survey is mainly on testing properties of functions (that is, testing for membership in various function classes). Though the perspective taken in the current monograph is different, there are naturally several results that appear in both articles, possibly with different levels of detail.

For the broader context of sublinear-time approximation algorithms see [104, 47]. For a survey on *Streaming* (where the constraint is sublinear *space* rather than time), see [107].

References

[1] N. Ailon and B. Chazelle, "Information theory in property testing and monotonicity testing in higher dimensions," *Information and Computation*, vol. 204, pp. 1704–1717, 2006.

[2] N. Ailon, B. Chazelle, S. Comandur, and D. Liue, "Estimating the distance to a monotone function," in *Proceedings of the Eight International Workshop on Randomization and Computation (RANDOM)*, pp. 229–236, 2004.

[3] N. Alon, "Testing subgraphs of large graphs," *Random Structures and Algorithms*, vol. 21, pp. 359–370, 2002.

[4] N. Alon, A. Andoni, T. Kaufman, K. Matulef, R. Rubinfeld, and N. Xie, "Testing k-wise and almost k-wise independence," in *Proceedings of the Thirty-Ninth Annual ACM Symposium on the Theory of Computing*, pp. 496–505, 2007.

[5] N. Alon, S. Dar, M. Parnas, and D. Ron, "Testing of clustering," *SIAM Journal on Discrete Math*, vol. 16, no. 3, pp. 393–417, 2003.

[6] N. Alon, R. A. Duke, H. Lefmann, V. Rodl, and R. Yuster, "The algorithmic aspects of the regularity lemma," *Journal of Algorithms*, vol. 16, pp. 80–109, 1994.

[7] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy, "Efficient testing of large graphs," *Combinatorica*, vol. 20, pp. 451–476, 2000.

[8] N. Alon, E. Fischer, I. Newman, and A. Shapira, "A combinatorial characterization of the testable graph properties: It's all about regularity," in *Proceedings of the Thirty-Eighth Annual ACM Symposium on the Theory of Computing*, pp. 251–260, 2006.

[9] N. Alon, T. Kaufman, M. Krivelevich, and D. Ron, "Testing triangle freeness in general graphs," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 279–288, 2006.

[10] N. Alon and M. Krivelevich, "Testing k-colorability," *SIAM Journal on Discrete Math*, vol. 15, no. 2, pp. 211–227, 2002.

[11] N. Alon, M. Krivelevich, T. Kaufman, S. Litsyn, and D. Ron, "Testing Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 4032–4038, 2005. An extended abstract of this paper appeared under the title: Testing Low-Degree Polynomials over GF(2), in the proceedings of RANDOM 2003.

[12] N. Alon, M. Krivelevich, I. Newman, and M. Szegedy, "Regular languages are testable with a constant number of queries," *SIAM Journal on Computing*, pp. 1842–1862, 2001.

[13] N. Alon, P. D. Seymour, and R. Thomas, "A separator theorem for graphs with an excluded minor and its applications," in *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing (STOC)*, pp. 293–299, 1990.

[14] N. Alon and A. Shapira, "Testing satisfiability," *Journal of Algorithms*, vol. 47, pp. 87–103, 2003.

[15] N. Alon and A. Shapira, "Testing subgraphs in directed graphs," *Journal of Computer and System Sciences*, vol. 69, pp. 354–482, 2004.

[16] N. Alon and A. Shapira, "A characterization of the (natural) graph properties testable with one-sided error," in *Proceedings of the Forty-Sixth Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 429–438, 2005.

[17] N. Alon and A. Shapira, "Every monotone graph property is testable," in *Proceedings of the Thirty-Seventh Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 129–137, 2005. To appear in SICOMP.

[18] N. Alon and A. Shapira, "A characterization of easily testable induced subgraphs," *Combinatorics Probability and Computing*, vol. 15, pp. 791–805, 2006.

[19] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," *Journal of the ACM*, vol. 45, no. 1, pp. 501–555, 1998. a preliminary version appeard in Proc. 33^{rd} FOCS, 1992.

[20] S. Arora and M. Sudan, "Improved low-degree testing and its applications," in *Proceedings of the Thirty-Second Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 485–495, 1997.

[21] L. Babai, L. Fortnow, L. Levin, and M. Szegedy, "Checking computations in polylogarithmic time," in *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing (STOC)*, pp. 21–31, 1991.

[22] L. Babai, L. Fortnow, and C. Lund, "Non-deterministic exponential time has two-prover interactive protocols," *Computational Complexity*, vol. 1, no. 1, pp. 3–40, 1991.

[23] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld, "The complexity of approximating the entropy," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 132–150, 2005.

[24] T. Batu, F. Ergun, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld, and R. Sami, "A sublinear algorithm for weakly approximating edit distance," in *Proceedings of the Thirty-Fifth Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 316–324, 2003.

[25] T. Batu, E. Fischer, L. Fortnow, R. Kumar, and R. Rubinfeld, "Testing random variables for independence and identity," in *Proceedings of the Forty-Second Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 442–451, 2001.

[26] T. Batu, L. Fortnow, R. Rubinfeld, W. Smith, and P. White, "Testing that distributions are close," in *Proceedings of the Forty-First Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 259–269, 2000.

[27] T. Batu, R. Kumar, and R. Rubinfeld, "Sublinear algorithms for testing monotone and unimodal distributions," in *Proceedings of the Thirty-Sixth Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 381–390, 2004.

[28] T. Batu, R. Rubinfeld, and P. White, "Fast approximate PCPs for multidimensional bin-packing problems," *Information and Computation*, vol. 196, no. 1, pp. 42–56, 2005.

[29] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan, "Linearity testing over characteristic two," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1781–1795, 1996.

[30] M. Bellare, O. Goldreich, and M. Sudan, "Free bits, PCPs and nonapproximability – towards tight results," *SIAM Journal on Computing*, vol. 27, no. 3, pp. 804–915, 1998.

[31] I. Ben-Eliezer, T. Kaufman, M. Krivelevich, and D. Ron, "Comparing the strength of query types in property testing: the case of testing k-colorability," in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

[32] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova, "3CNF properties are hard to test," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 1–21, 2005.

[33] I. Benjamini, O. Schramm, and A. Shapira, "Every minor-closed property of sparse graphs is testable," in *Proceedings of the Fourtieth Annual ACM Symposium on the Theory of Computing*, pp. 393–402, 2008.

[34] E. Blais, "Testing juntas almost optimally," in *Proceedings of the Fourty-First Annual ACM Symposium on the Theory of Computing*, pp. 151–158, 2009.

[35] M. Blum, M. Luby, and R. Rubinfeld, "Self-testing/correcting with applications to numerical problems," *Journal of the ACM*, vol. 47, pp. 549–595, 1993.

[36] A. Bogdanov, K. Obata, and L. Trevisan, "A lower bound for testing 3-colorability in bounded-degree graphs," in *Proceedings of the Forty-Third Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 93–102, 2002.

[37] A. Bogdanov and L. Trevisan, "Lower bounds for testing bipartiteness in dense graphs," in *Proceedings of the Nineteenth Computationa Complexity Conference (CCC)*, 2004.

[38] B. Bollig, "Property testing and the branching program size," in *Proceedings of FCT, Lecture notes in Computer Science 3623*, pp. 258–269, 2005.

[39] B. Bollig and I. Wegener, "Functions that have read-once branching programs of quadratic size are not necessarily testable," *Information Processing Letters*, vol. 87, no. 1, pp. 25–29, 2003.

[40] C. Borgs, J. Chayes, L. Lovász, V. T. Sós, B. Szegedy, and K. Vesztergombi, "Graph limits and parameter testing," in *Proceedings of the Thirty-Eighth Annual ACM Symposium on the Theory of Computing*, pp. 261–270, 2006.

[41] S. Chakraborty, E. Fischer, O. Lachish, A. Matsliah, and I. Newman, "Testing *st*-connectivity," in *Proceedings of the Eleventh International Workshop on Randomization and Computation (RANDOM)*, pp. 380–394, 2007.

[42] B. Chazelle, R. Rubinfeld, and L. Trevisan, "Approximating the minimum spanning tree weight in sublinear time," in *Automata, Languages and Programming: Twenty-Eighth International Colloquium (ICALP)*, pp. 190–200, 2001.

[43] H. Chockler and D. Gutfreund, "A lower bound for testing juntas," *Information Processing Letters*, vol. 90, no. 6, pp. 301–305, 2006.

[44] A. Czumaj, A. Shapira, and C. Sohler, "Testing hereditary properties of non-expanding bounded-degree graphs," Technical Report TR07-083, Electronic Colloquium on Computational Complexity (ECCC), 2007.

[45] A. Czumaj and C. Sohler, "Property testing with geometric queries," in *Proceedings of the Ninth Annual European Symposium on Algorithms (ESA)*, pp. 266–277, 2001.

[46] A. Czumaj and C. Sohler, "Abstract combinatorial programs and efficient property testers," *SIAM Journal on Computing*, vol. 34, no. 3, pp. 580–615, 2005.

[47] A. Czumaj and C. Sohler, "Sublinear-time algorithms," *Bulletin of the EATCS*, vol. 89, pp. 23–47, 2006.

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

[49] A. Czumaj, C. Sohler, and M. Ziegler, "Property testing in computation geometry," in *Proceedings of the Eighth Annual European Symposium on Algorithms (ESA)*, pp. 155–166, 2000.

[50] I. Diakonikolas, H. K. Lee, K. Matulef, K. Onak, R. Rubinfeld, R. A. Servedio, and A. Wan, "Testing for concise representations," in *Proceedings of the Forty-Eighth Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 549–557, 2007.

[51] I. Diakonikolas, H. K. Lee, K. Matulef, R. A. Servedio, and A. Wan, "Efficient testing of sparse GF(2) polynomials," in *Automata, Languages and Programming: Thirty-Fifth International Colloquium (ICALP)*, pp. 502–514, 2008.

[52] E. A. Dinic, A. V. Karazanov, and M. V. Lomonosov, "On the structure of the system of minimum edge cuts in a graph," *Studies in Discrete Optimizations*, pp. 290–306, 1976. In Russian.

[53] Y. Dinitz and J. Westbrook, "Maintaining the classes of 4-edge-connectivity in a graph on-line," *Algorithmica*, vol. 20, no. 3, pp. 242–276, 1998.

[54] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky, "Improved testing algorithms for monotonocity," in

Proceedings of the Third International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM), pp. 97–108, 1999.

[55] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan, "Spot-checkers," *Journal of Computer and System Sciences*, vol. 60, no. 3, pp. 717–751, 2000.

[56] S. Fattal and D. Ron, "Approximating the distance to monotonicity in high dimensions," To appear in *Transactions on Algorithms*, 2009.

[57] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, "Approximating clique is almost NP-complete," *Journal of the ACM*, pp. 268–292, 1996.

[58] E. Fischer, "The art of uninformed decisions: A primer to property testing," *Bulletin of the European Association for Theoretical Computer Science*, vol. 75, pp. 97–126, 2001.

[59] E. Fischer, "Testing graphs for colorability properties," *Random Structures and Algorithms*, vol. 26, no. 3, pp. 289–309, 2005.

[60] E. Fischer and L. Fortnow, "Tolerant versus intolerant testing for boolean properties," *Theory of Computing*, vol. 2, pp. 173–183, 2006.

[61] E. Fischer, G. Kindler, D. Ron, S. Safra, and S. Samorodnitsky, "Testing juntas," *Journal of Computer and System Sciences*, vol. 68, no. 4, pp. 753–787, 2004.

[62] E. Fischer, O. Lachish, A. Matsliah, I. Newman, and O. Yahalom, "On the query complexity of testing orientations for being Eulerian," in *Proceedings of the Twelveth International Workshop on Randomization and Computation (RANDOM)*, pp. 402–415, 2008.

[63] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samrodnitsky, "Monotonicity testing over general poset domains," in *Proceedings of the Thirty-Fourth Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 474–483, 2002.

[64] E. Fischer and I. Newman, "Testing versus estimation of graph properties," *SIAM Journal on Computing*, vol. 37, no. 2, pp. 482–501, 2007.

[65] E. Fischer, I. Newman, and J. Sgall, "Functions that have read-twice constant width branching programs are not necessarily testable," *Random Structures and Algorithms*, vol. 24, no. 2, pp. 175–193, 2004.

[66] K. Friedl and M. Sudan, "Some improvements to total degree tests," in *Proceedings of the 3rd Annual Israel Symposium on Theory of Computing and Systems*, pp. 190–198, 1995. Corrected version available online at http://theory.lcs.mit.edu/~madhu/papers/friedl.ps.

[67] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson, "Self testing/correcting for polynomials and for approximate functions," in *Proceedings of the Thirty-Second Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 32–42, 1991.

[68] D. Glasner and R. A. Servedio, "Distribution-free testing lower bounds for basic Boolean functions," in *Proceedings of the Eleventh International Workshop on Randomization and Computation (RANDOM)*, pp. 494–508, 2007.

[69] O. Goldreich, "Combinatorial property testing - a survey," in *Randomization Methods in Algorithm Design*, pp. 45–60, 1998.

[70] O. Goldreich, "Short locally testable codes and proofs (a survey)," Technical Report TR05-014, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[71] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samordinsky, "Testing monotonicity," *Combinatorica*, vol. 20, no. 3, pp. 301–337, 2000.

[72] O. Goldreich, S. Goldwasser, and D. Ron, "Property testing and its connection to learning and approximation," *Journal of the ACM*, vol. 45, no. 4, pp. 653–750, 1998.

[73] O. Goldreich and D. Ron, "Property testing in bounded degree graphs," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 406–415, 1997. This is an extended abstract of [76] which includes algorithms for testing 2-vertex connectivity and 3-vertex connectivity.

[74] O. Goldreich and D. Ron, "A sublinear bipartite tester for bounded degree graphs," *Combinatorica*, vol. 19, no. 3, pp. 335–373, 1999.

[75] O. Goldreich and D. Ron, "On testing expansion in bounded-degree graphs," *Electronic Colloqium on Computational Complexity*, vol. 7, no. 20, 2000.

[76] O. Goldreich and D. Ron, "Property testing in bounded degree graphs," *Algorithmica*, pp. 302–343, 2002.

[77] O. Goldreich and D. Ron, "Algorithmic aspects of property testing in the dense graphs model," in *Proceedings of the Thirteenth International Workshop on Randomization and Computation (RANDOM)*, pp. 520–533, 2009.

[78] O. Goldreich and L. Trevisan, "Three theorems regarding testing graph properties," *Random Structures and Algorithms*, vol. 23, no. 1, pp. 23–57, 2003.

[79] M. Gonen and D. Ron, "On the benefits of adaptivity in property testing of dense graphs," in *Proceedings of the Eleventh International Workshop on Randomization and Computation (RANDOM)*, pp. 525–537, 2007. To appear in *Algorithmica*.

[80] V. Guruswami and A. Rudra, "Tolerant locally testable codes," in *Proceedings of the Ninth International Workshop on Randomization and Computation (RANDOM)*, pp. 306–317, 2005.

[81] S. Halevy and E. Kushilevitz, "Distribution-free property testing," in *Proceedings of the Seventh International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pp. 341–353, 2003.

[82] S. Halevy and E. Kushilevitz, "Distribution-free connectivity testing," in *Proceedings of the Eight International Workshop on Randomization and Computation (RANDOM)*, pp. 393–404, 2004.

[83] S. Halevy and E. Kushilevitz, "A lower bound for distribution-free monotonicity testing," in *Proceedings of the Ninth International Workshop on Randomization and Computation (RANDOM)*, pp. 330–341, 2005.

[84] S. Halevy and E. Kushilevitz, "Distribution-free property testing," *SIAM Journal on Computing*, vol. 37, no. 4, pp. 1107–1138, 2007.

[85] S. Halevy, O. Lachish, I. Newman, and D. Tsur, "Testing orientation properties," Technical Report TR05-153, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[86] S. Halevy, O. Lachish, I. Newman, and D. Tsur, "Testing properties of constraint-graphs," in *Proceedings of the Twenty-Second IEEE Annual Conference on Computational Complexity (CCC)*, pp. 264–277, 2007.

[87] A. Hassidim, J. Kelner, H. Nguyen, and K. Onak, "Local graph partitions for approximation and testing," in *Proceedings of the Fiftieth Annual Symposium on Foundations of Computer Science (FOCS)*, 2009.

[88] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$," *Acta Mathematica*, vol. 182, pp. 105–142, 1999.

[89] D. S. Hochbaum and D. B. Shmoys, "Using dual approximation algorithms for scheduling problems: Theoretical and practical results," *Journal of the ACM*, vol. 34, pp. 144–162, January 1987.

[90] D. S. Hochbaum and D. B. Shmoys, "A polynomial approximation scheme for machine scheduling on uniform processors: Using the dual approximation approach," *SIAM Journal on Computing*, vol. 17, no. 3, pp. 539–551, 1988.

[91] H. Ito and Y. Yoshida, "Property testing on k-vertex connectivity of graphs," in *Automata, Languages and Programming: Thirty-Fifth International Colloquium (ICALP)*, pp. 539–550, 2008.

[92] C. S. Jutla, A. C. Patthak, A. Rudra, and D. Zuckerman, "Testing low-degree polynomials over prime fields," in *Proceedings of the Forty-Fifth Annual Symposium on Foundations of Computer Science (FOCS)*, 2004.

[93] S. Kale and C. Seshadhri, "Testing expansion in bounded degree graphs," Technical Report TR07-076, Electronic Colloquium on Computational Complexity (ECCC), 2007.

[94] S. Kale and C. Seshadhri, "Testing expansion in bounded degree graphs," in *Automata, Languages and Programming: Thirty-Fifth International Colloquium (ICALP)*, pp. 527–538, 2008.

[95] D. Karger, "Global min-cuts in $\mathcal{RNC}$ and other ramifications of a simple min-cut algorithm," in *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 21–30, 1993.

[96] T. Kaufman, M. Krivelevich, and D. Ron, "Tight bounds for testing bipartiteness in general graphs," *SIAM Journal on Computing*, vol. 33, no. 6, pp. 1441–1483, 2004.

[97] T. Kaufman and S. Litsyn, "Almost orthogonal linear codes are locally testable," in *Proceedings of the Forty-Sixth Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 317–326, 2005.

[98] T. Kaufman and D. Ron, "Testing polynomials over general fields," *SIAM Journal on Computing*, vol. 35, no. 3, pp. 779–802, 2006.

[99] T. Kaufman and M. Sudan, "Algebraic property testing: The role of invariance," in *Proceedings of the Fourtieth Annual ACM Symposium on the Theory of Computing*, pp. 403–412, 2008. For a full version see the ECCC technical report TR07–111.

[100] M. Kearns and D. Ron, "Testing problems with sub-learning sample complexity," *Journal of Computer and System Sciences*, vol. 61, no. 3, pp. 428–456, 2000.

[101] J. C. Kiefer, *Introduction to Statistical Inference*. Springer Verlag, 1987.

[102] Y. Kohayakawa, B. Nagle, and V. Rödl, "Efficient testing of hypergraphs," in *Automata, Languages and Programming: Twenty-Ninth International Colloquium (ICALP)*, pp. 1017–1028, 2002.

[103] S. Kopparty and S. Saraf, "Tolerant linearity testing and locally testable codes," in *Proceedings of the Thirteenth International Workshop on Randomization and Computation (RANDOM)*, pp. 601–614, 2009.

[104] R. Kumar and R. Rubinfeld, "Sublinear time algorithms," Samir Khuller's SIGACT News column, 2003.

[105] S. Marko and D. Ron, "Distance approximation in bounded-degree and general sparse graphs," *Transactions on Algorithms*, vol. 5, no. 2, 2009. Article number 22.

[106] M. Mihail, "Conductance and convergence of Markov chains — A combinatorial treatment of expanders," in *Proceedings of the Thirtieth Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 526–531, 1989.

[107] S. Muthukrishnan, "Data streams: Algorithms and applications," *Foundations and Trends in Theoretical Computer Science*, vol. 1, no. 2, 2005.

[108] A. Nachmias and A. Shapira, "Testing the expansion of a graph," Technical Report TR07-118, Electronic Colloquium on Computational Complexity (ECCC), 2007.

[109] D. Naor, D. Gusfield, and C. Martel, "A fast algorithm for optimally increasing the edge-connectivity," *SIAM Journal on Computing*, vol. 26, no. 4, pp. 1139–1165, 1997.

[110] I. Newman, "Testing membership in languages that have small width branching programs," *SIAM Journal on Computing*, vol. 31, no. 5, pp. 1557–1570, 2002.

[111] Y. Orenstein, "Private communication," To appear in Y. Orenstein's MSC thesis, 2009.

[112] M. Parnas and D. Ron, "Testing the diameter of graphs," *Random Structures and Algorithms*, vol. 20, no. 2, pp. 165–183, 2002.

[113] M. Parnas and D. Ron, "Testing metric properties," *Information and Computation*, vol. 187, no. 2, pp. 155–195, 2003.

[114] M. Parnas, D. Ron, and R. Rubinfeld, "Testing membership in parenthesis languages," *Random Structures and Algorithms*, vol. 22, no. 1, pp. 98–138, 2003.

[115] M. Parnas, D. Ron, and R. Rubinfeld, "Tolerant property testing and distance approximation," *Journal of Computer and System Sciences*, vol. 72, no. 6, pp. 1012–1042, 2006.

[116] M. Parnas, D. Ron, and A. Samorodnitsky, "Testing basic boolean formulae," *SIAM Journal on Discrete Math*, vol. 16, no. 1, pp. 20–46, 2002.

[117] S. Raskhodnikova, D. Ron, R. Rubinfeld, and A. Smith, "Strong lower bonds for approximating distributions support size and the distinct elements problem," in *Proceedings of the Forty-Eighth Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 559–568, 2007. To appear in *SIAM Journal on Computing*.

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

[119] N. Robertson and P. D. Seymour, "Graph minors. XX. Wagner's conjecture," *Journal of Combinatorial Theory Series B*, vol. 92, no. 2, pp. 325–357, 2004.

[120] D. Ron, "Property testing," in *Handbook on Randomization, Volume II*, (S. Rajasekaran, P. M. Pardalos, J. H. Reif, and J. D. P. Rolim, eds.), pp. 597–649, 2001.

[121] D. Ron, "Property testing: A learning theory perspective," *Foundations and Trends in Machine Learning*, vol. 1, no. 3, pp. 307–402, 2008.

[122] R. Rubinfeld and R. Servedio, "Testing monotone high-dimensional distributions," in *Proceedings of the Thirty-Seventh Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 147–156, 2005.

[123] R. Rubinfeld and M. Sudan, "Robust characterization of polynomials with applications to program testing," *SIAM Journal on Computing*, vol. 25, no. 2, pp. 252–271, 1996.

[124] E. Szemerédi, "Regular partitions of graphs," in *Proceedings, Colloque Inter. CNRS*, pp. 399–401, 1978.

[125] L. G. Valiant, "A theory of the learnable," *CACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[126] P. Valiant, "Testing symmetric properties of distributions," in *Proceedings of the Fourtieth Annual ACM Symposium on the Theory of Computing*, pp. 383–392, 2008.