Lecture Notes in Computer Science 1001
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Advisory Board: W. Brauer D. Gries J. Stoer

Madhu Sudan

Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems

Springer

Series Editors

Gerhard Goos
Universität Karlsruhe
Vincenz-Priessnitz-Straße 3, D-76128 Karlsruhe, Germany

Juris Hartmanis
Department of Computer Science, Cornell University
4130 Upson Hall, Ithaca, NY 14853, USA

Jan van Leeuwen
Department of Computer Science,Utrecht University
Padualaan 14, 3584 CH Utrecht, The Netherlands

Author

Madhu Sudan
IBM Thomas J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598, USA

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Sudan, Madhu:
Efficient checking of polynomials and proofs and the hardness of approximation problems / Madhu Sudan. - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1996
(Lecture notes in computer science ; 1001)
ISBN 3-540-60615-7

NE: GT

CR Subject Classification (1991): F.2, F.3.1, D.2.5-6, E.4, F.4.1, G.1, G.3, I.1.2

ISBN 3-540-60615-7 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10512261 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Foreword

How difficult is it to compute an approximate solution to an NP-optimization problem? The central importance of this issue has been recognized since the early 1970s, when Cook and Karp formulated the theory of NP-hard, and therefore computationally intractable unless P = NP, problems. To sidestep this difficulty, researchers asked whether there are polynomial time algorithms for producing near-optimal solutions to these optimization problems. This approach was successful for some problems such as bin packing, but other problems such as the Euclidean traveling salesman problem and max-clique resisted all efforts at the design of efficient approximation algorithms. Sudan's dissertation describes a general technique, akin to NP-completeness, for establishing the computational intractability of approximation problems (under the assumption that P $\neq$ NP). The dissertation establishes approximation hardness for all complete problems in the complexity class max-SNP: this includes basic problems such as the Euclidean traveling salesman problem, max-2SAT, and Euclidean Steiner tree. Elsewhere, these techniques have other important problems such as chromatic number, set cover, and shortest vector in a lattice. There is little doubt that the new techniques are very generally applicable, and are fundamental to establishing the intractability of approximate solutions to NP-optimization problems.

The techniques themselves are interesting and deep. They build upon a sequence of beautiful previous results on probabilistically checkable proofs. Sudan's dissertation provides a new charaterization of the complexity class NP, of languages such that membership of a string x in the language can be established by a polynomial size proof. The new characterization shows that the proofs of membership can be made surprisingly robust: the robust proofs are still polynomially long, but can be checked (in a probabilistic sense) by probing only a constant number of randomly chosen bits of the proof. The proof of this theorem is a technical tour de force; it has several major new ingredients in addition to masterfully building upon the previous work of Babai et al., Feige et al., and Arora and Safra, to name a few.

One new ingredient is a beautiful technique for creating long but very robust proofs based on self-correction properties of linear functions. Another is a new *low-degree test* that probes the value of a multivariate function at only a constant number of points and verifies whether it is close to some low-degree polynomial. The dissertation also introduces a new connection between robust probabilistically checkable proofs and the approximation hardness of the optimization problem max-SAT. This connection is the basis of the new technique for proving approximation hardness of NP optimization problems.

Sudan's dissertation introduces a new framework for the general algebraic problem of efficiently reconstructing a low degree multivariate polynomial from erroneous data. Using this framework, it presents self-contained proofs of several previous results on probabilistically checkable proofs, as well as the new results. In this framework, the connection of this work to coding theory becomes more explicit as well; the testers and correctors for multivariate polynomials developed in the dissertation yield codes with very efficient error-detection and error-correction schemes.

The work reported in this dissertation has already had, and will continue to have, a profound influence on theoretical computer science.

November 1995

Umesh Vazirani
Professor of Computer Science
University of California at Berkeley

Preface

The definition of the class NP (Cook [41], Levin [86]) highlights the problem of verification of proofs as one of central interest to theoretical computer science. Recent efforts have shown that the efficiency of the verification can be greatly improved by allowing the verifier access to random bits and accepting probabilistic guarantees from the verifier [20, 19, 50, 6]. We improve upon the efficiency of the proof systems developed above and obtain proofs which can be verified probabilistically by examining only a constant number of (randomly chosen) bits of the proof.

The efficiently verifiable proofs constructed here rely on the structural properties of low-degree polynomials. We explore the properties of these functions by examining some simple and basic questions about them. We consider questions of the form:

(testing) Given an oracle for a function f, is f close to a low-degree polynomial?

(correcting) Given an oracle for a function f that is close to a low-degree polynomial g, is it possible to efficiently reconstruct the value of g on any given input using an oracle for f?

These questions have been raised before in the context of coding theory as the problems of error-detecting and error-correcting of codes. More recently, interest in such questions has revived due to their connection with the area of program result checking. We use results from coding theory as a starting point and combine these with several algorithmic techniques including pairwise independent sampling to give efficient randomized algorithms for these tasks. As a consequence we obtain fast randomized algorithms for error-detection and error-correction for some well-known codes.

The expressive nature of low-degree polynomials suffices to capture the complexity of the class NP, and we translate our results on the efficiency of the testing and correcting procedures into two different efficiently verifiable proof systems for deciding membership questions for NP languages. One proof system generates small and somewhat efficiently verifiable proofs, and the other generates very large but very efficiently verifiable proofs. We then employ new techniques from the work of Arora and Safra [6] to compose these proof systems to obtain small proofs that can be verified by probing them in just a constant number of (randomly chosen) bits.

An important consequence of this result is that for a large variety of NP-complete optimization problems, it can be shown that finding even approximate solutions is an NP-hard problem. The particular class of optimization problems we consider is MAX SNP, introduced by Papadimitriou and Yannakakis [93]. For every MAX SNP-hard problem we show that there is a constant ϵ, such that approximating the optimum to within a relative error of ϵ is NP-hard.

This version. This version of the dissertation is essentially the same as the one filed at the University of California at Berkeley in 1992. A few proofs have been fixed to address the comments of several readers who pointed out errors in the earlier version. In addition this version has an addendum at the end of every chapter to bring the reader up to date with the various developments in the subjects covered in this thesis during the period from mid-1992 to mid-1995.

Acknowledgments

I am greatly indebted to Umesh Vazirani, my advisor, for the five years of careful nurturing that he has given me. His highly intuitive and creative approach to technical issues has left a deep impact on me. His ability to throw new light onto existing ideas have been a big factor in aiding my thinking. I owe a lot to Umesh, most importantly his taste and appreciation of mathematics, some of which has hopefully rubbed off on me. Umesh was more than just a technical adviser, and many are the times I have sought his counsel on non-technical matters, and I am thankful to him for his patience with me.

I enjoyed the numerous meetings I have had with Dick Karp, during which he monitored my progress, bolstered my confidence and at the same time provided me with his candid opinion on my work. Every meeting with Dick was a highly fulfilling experience. I'd also like to thank him for the wonderful courses he taught us, which were a great part of my learning experience at Berkeley.

A large portion of the work done here has been motivated by the work of Manuel Blum, and meetings with him over the last summer provided a turning point in my research. His enthusiasm for our work proved to be a much needed catalyst, and his pointers, which led us to the wonderful world of coding theory, were crucial to some of the results described here. Mike Luby has been a constant source of inspiration to me. He has been very generous with his time, during which I gleaned immense technical knowledge from him. His study groups at ICSI have been among my best sources of information and are largely responsible for providing me with a sound technical footing. I would like to thank Dorit Hochbaum for the many hours she spent with me in my last few months here and for sharing her wealth of knowledge on approximation algorithms with me. I would also like to thank Sachin Maheshwari at the Indian Institute of Technology at New Delhi, whose enthusiasm for the area of theoretical computer science is perhaps the single largest factor for my working in this area today.

I have been very fortunate to have found a large set of great people to work with. Ronitt Rubinfeld introduced me to the area of program checking, and much of my initial as well as current work has been done jointly with her. I am grateful to her for having shared her ideas with me, for the respect

she showed me, and for encouraging me to work in this area. The results of Chapters 2 and 3 were obtained jointly with Ronitt and Peter Gemmell. I'd like to thank Rajeev Motwani for playing the role of my mentor and for passing on his keen quest for knowledge, which provided the motivation behind much of the work of Chapters 4 and 5. The work described in these chapters was done jointly with Rajeev, Sanjeev Arora, Carsten Lund and Mario Szegedy. Sanjeev deserves a special thanks for having provided me with wonderful explanations of the work he was involved in and for bringing me up to date in his area.

The academic environment at Berkeley has on the one hand been a very stimulating one and on the other it has provided me with a wonderful set of friends. Milena Mihail has been a true friend, whose advice I could always count on. I learnt much from her in my early years here. Abhijit Sahay has always provided me with a great sounding board for my ideas, as I hope I have for him. His company extended well beyond the office we shared, and, along with Savita and Usha, he has been the closest I had to a family of my own in Berkeley. I was also fortunate to have found such patient officemates in Jim Ruppert and Yiannis Emiris, who never seemed to tire of me. Thanks especially to Diane Hernek, whose company was always a pleasure. Studying at Berkeley has been a fun-filled experience and I'd like to thank Will Evans, Sridhar Rajagopalan, Sigal Ar, Dana Randall, and Z Sweedyk for this.

I was fortunate to have a company at home as stimulating as at the office: Sushil Verma, my housemate for over three years, was a great friend who always provided a willing ear to the problems I was working on. Pratap Khedkar was an infinite source of information to be tapped, and many are the times when his information helped me in my work by either setting me on the right track or stopping me from spending time on dead ends. I'd like to thank K.J. Singh, Narendra Shenoy, Audumbar Padgaonkar, Anant Jhingran, Sharad Malik, Savita Sahay, Diane Bailey, Sampath Vedant, Huzur Saran, and Rajiv Murgai for all the wonderful times.

This thesis had the (mis)fortune of being read by a lot more readers than originally planned for – and consequently many mistakes were brought out from the earlier version. I thank Mihir Bellare, Michael Goldman, Oded Goldreich, Shafi Goldwasser, Jaikumar Radhakrishnan, and Karl-Heinz Schmidt for their comments on this thesis and related issues.

Last, I would like to thank my family members – my parents and my sister Satya who have been most loving and understanding to me especially during my long absence from home; and Madhulika for entering my life. Amma and Appa, this thesis is dedicated to you.

Table of Contents