Preface

Proof Verification and Approximation Algorithms. – Hardly any area in theoretical computer science has been more lively and flourishing during the last few years. Different lines of research which had been developed independently of each other over the years culminated in a new and unexpected characterization of the well-known complexity class $\mathcal{NP}$, based on probabilistically checking certain kinds of proofs. This characterization not only sheds new light on the class $\mathcal{NP}$ itself, it also allows proof of non-approximability results for optimization problems which, for a long time, had seemed to be out of reach. This connection, in turn, has motivated scientists to take a new look at approximating $\mathcal{NP}$-hard problems as well – with quite surprising success. And apparently, these exciting developments are far from being finished.

We therefore judged "Proof Verification and Approximation Algorithms" an ideal topic for the first in a new series of research seminars for young scientists, to be held at the International Conference and Research Center for Computer Science at Schloß Dagstuhl in Germany. This new series of seminars was established by the German Society for Computer Science (Gesellschaft für Informatik, GI) with the aim of introducing students and young scientists to important new research areas and results not yet accessible in text books or covered in the literature in a comprehensive way.

When we announced our seminar we encountered considerable interest and received numerous responses. We were able to select 21 qualified doctoral students and postdocs. Each participant then was requested to give a lecture, usually based on several research articles or technical reports, and to submit, in preliminary form and before the workshop began, an exposition of the topic assigned to him/her. The actual workshop then took place April 21–25, 1997 at Schloß Dagstuhl. All participants were very well prepared and highly motivated. We heard excellent talks and had many interesting and stimulating discussions, in the regular sessions as well as over coffee or some enlightening glass of wine after dinner.

This volume contains revised versions of the papers submitted by the participants. The process of revision involved, among other things, unifying notation, removing overlapping parts, adding missing links, and even combining some of the papers into single chapters. The resulting text should now be a coherent and essentially self-contained presentation of the enormous recent progress facilitated by the interplay between the theory of probabilistically checkable proofs and approximation algorithms. While it is certainly not a textbook in the usual sense, we nevertheless believe that it can be helpful for all those who are just starting out to learn about these subjects, and hopefully even to those looking for a coherent treatment of the subject for teaching purposes.

Our workshop was sponsored generously by Special Interest Group 0 (Fachbereich "Grundlagen der Informatik") of the German Society for Computer Science (GI) and by the International Conference and Research Center for Computer Science (Internationales Begegnungs- und Forschungszentrum für Informatik, IBFI) at Schloß Dagstuhl. We owe them and the staff at Schloß Dagstuhl many thanks for a very successful and enjoyable meeting.

München, Berlin
September 1997

Ernst W. Mayr
Hans Jürgen Prömel
Angelika Steger

Prologue

Exam time. Assume you are the teaching assistant for some basic course with s students, s very large. The setup for the exam is as follows:

(1) The exam consists of q yes/no questions.

(2) A student passes *if and only if* he or she answers all questions correctly.

You assume that, on average, you'll need at least half a second to check the correctness of each answer. Since you expect the number of students to be close to one thousand (it is a very popular basic course!) and since the number of questions will be several hundred a rough estimate shows that you are going to spend almost a whole week grading the exam. Ooff.

Is there a faster way?

Certainly not in general: in the worst case you really might have to look at all $s \cdot q$ answers in order to rule out a false decision. But what if we relax the second condition slightly and replace it by

(2') A student definitely passes the exam if he or she answers all questions correctly. A student who does not answer all questions correctly may pass only with a small probability, say $\leqslant 10^{-3}$, independently of the answers he or she gives.

Now you suddenly realize that the grading can actually be done in about $45s$ seconds, even regardless of the actual number q of questions asked in the exam. That is, a single day should suffice. Not too bad.

How is this possible? Find out by reading this book! And enjoy!

Introduction

During the last few years we have seen quite spectacular progress in the area of approximation algorithms. For several fundamental optimization problems we now actually know matching upper and lower bounds for their approximability (by polynomial time algorithms).

Perhaps surprisingly, it turned out that for several of these problems, including the well-known MAX3SAT, SETCOVER, MAXCLIQUE, and CHROMATICNUMBER, rather simple and straightforward algorithms already yield the essentially best possible bound, at least under some widely believed assumptions from complexity theory. The missing step for tightening the gap between upper and lower bound was the improvement of the lower or non-approximability bound. Here the progress was initiated by a result in a seemingly unrelated area, namely a new characterization of the well-known complexity class $\mathcal{NP}$. This result is due to Arora, Lund, Motwani, Sudan, and Szegedy and is based on so-called probabilistically checkable proofs. While already very surprising and certainly interesting by itself, this result has given rise to fairly general techniques for deriving non-approximability results, and it initiated a large amount of subsequent work.

On the other hand, as if this so-to-speak "negative" progress had inspired the research community, the last few years have also brought us considerable progress on the "positive" or algorithmic side. Perhaps the two most spectacular results in this category are the approximation of MAXCUT using semidefinite programming, by Goemans and Williamson, and the development of polynomial time approximation schemes for various geometric problems, obtained independently by Arora and Mitchell.

These notes give an essentially self-contained exposition of some of these new and exciting developments for the interplay between complexity theory and approximation algorithms. The concepts, methods and results are presented in a unified way that should provide a smooth introduction to newcomers. In particular, we expect these notes to be a useful basis for an advanced course or reading group on probabilistically checkable proofs and approximability.

Overview and Organization of this Book

To be accessible for people from different backgrounds these notes start with three introductory chapters. The first chapter provides an introduction to the world of complexity theory and approximation algorithms, as needed for the subsequent treatment. While most of the notions and results from complexity theory that are introduced here are well-known and classical, the part on approximation algorithms incorporates some very recent results which in fact reshape a number of definitions and viewpoints. It also includes the proof by Trevisan [Tre97] that Max3Sat is $\mathcal{APX}$-complete.

The second chapter presents a short introduction to randomized algorithms, demonstrating their usefulness by showing that an essentially trivial randomized algorithm for MaxE3Sat (the version of Max3Sat in which all clauses have exactly three literals) has expected performance ratio 8/7. Later on, in Chapter 7, this ratio will be seen to be essentially best possible, assuming $\mathcal{P} \neq \mathcal{NP}$.

Concluding the introductory part, the third chapter describes various facets and techniques of derandomization, a term coined for the process of turning randomized algorithms into deterministic ones. Amongst other things in this chapter it is shown that the algorithm for MaxE3Sat is easily derandomized.

Chapters 4 to 10 are devoted to the concept of probabilistically checkable proofs and the implications for non-approximability. Chapter 4 introduces the so-called PCP-Theorem, a new characterization of $\mathcal{NP}$ in terms of probabilistically checkable proofs, and explains why and how they can be used to show non-approximability results. In particular, the nonexistence of polynomial time approximation schemes for $\mathcal{APX}$-complete problems and the non-approximability of MaxClique are shown in detail. A complete and self-contained proof of the PCP-Theorem is presented in Chapter 5. Chapter 6 is devoted to the so-called Parallel Repetition Theorem of Raz [Raz95] which is used heavily in subsequent chapters.

At the 1997 STOC, Håstad [Hås97b] presented an exciting paper showing that the simple algorithm of Chapter 2 for approximating MaxE3Sat is essentially best possible. Chapter 7 is devoted to this result of Håstad's. The chapter also introduces the concept of long codes and a method of analyzing these codes by means of discrete Fourier transforms. These tools will be reused later in Chapter 9.

Chapter 8 surveys the new reduction techniques for optimization problems using gadgets, a notion for the first time formally introduced within the framework of approximation algorithms by Bellare, Goldreich, and Sudan [BGS95].

MaxClique cannot be approximated up to a factor of $n^{1-\varepsilon}$ unless $\mathcal{NP} = \mathcal{ZPP}$. This result, also due to Håstad [Hås96a], is based on a version of the PCP-Theorem using so-called free bits. This concept, as well as Håstad's result, are described in Chapter 9.

As the final installment in this series of optimal non-approximability results, Chapter 10 presents the result of Feige [Fei96] stating that for SetCover the approximation factor of $\ln n$ achieved by a simple greedy algorithm is essentially best possible unless $\mathcal{NP} \subseteq \mathrm{DTIME}(n^{\mathcal{O}(\log\log n)})$.

The last three chapters of these notes, are devoted to new directions in the development of approximation algorithms. First, Chapter 11 surveys recent achievements in constructing approximation algorithms based on semidefinite programming. A generalization of linear programming, semidefinite programming had been studied before for some time and in various contexts. However, only a few years ago Goemans and Williamson [GW95] showed how to make use of it in order to provide good approximation algorithms for several optimization problems.

While the PCP-Theorem implied that no $\mathcal{APX}$-complete problem can have a polynomial time approximation scheme unless $\mathcal{NP} = \mathcal{P}$, it is quite surprising that many such problems nevertheless do have such approximation schemes when restricted to in a certain sense dense instances. Chapter 12 exemplifies a very general approach for such dense instances, due to Arora, Karger, and Karpinski [AKK95a].

The final chapter then presents one of the highlights of the work on approximation algorithms during recent years. It is the development of polynomial time approximation schemes for geometrical problems like the Euclidean traveling salesman problem, independently by Arora [Aro96, Aro97] and Mitchell [Mit96].

Notations and Conventions

Areas as lively and evolving as proof verification and approximation algorithms naturally do not have a standardized set of definitions and notations. Quite often the same phrase has a slightly different meaning in different papers, or different symbols have identical meaning. In these notes, we have striven for uniform notation and concepts. We have tried to avoid any redefinition of terms, and thus we sometimes had to choose between two (or more) equally well established alternatives (e.g., should approximation ratios be taken to always be $\geqslant 1$, or $\leqslant 1$, or depending on the type of approximation problem?).

We have also tried to avoid phrases like “reconsidering the proof of Theorem x in Chapter y we see that it also shows that ...”. Instead, we have attempted to prove all statements in the form in which they'll be needed later on. We hope that in this way we have been able to make the arguments easier to follow and to improve readability of the text.

Finally, we want to explicitly add some disclaimers and an apology. The intention of these notes certainly is *not* to present a survey, detailed or not, on the *history* of research in proof verification or approximation algorithms. This means in particular, that more often than not only the reference to a paper with the best bound or complexity is given, omitting an entire sequence of earlier work without which the final result would appear all but impossible. Of course, numerous citations and pointers to work that had a major impact in the field are given, but there are doubtlessly many omissions and erroneous judgments. We therefore would like to apologize to all those whose work does not receive proper credit in these notes.

Acknowledgments

This volume, and in particular its timely completion, has only been possible by the joint effort of all participants. We would like to thank them all. Editing this volume in its various stages has been a great pleasure. For the numerous technical details special thanks are due to Katja Wolf for taking care of the bibliography, Stefan Hougardy for preparing the index, and, last but not least, to Volker Heun, who really did an excellent job in solving the thousand and more remaining problems.