Statement of Research

Mikhail Alekhnovich

February 17, 2005

Abstract

I am mainly interested in theoretical aspects of Computer Science. The main focus of my research has been in propositional proof complexity and in computational complexity. My main research contributions are in applications of propositional proof techniques to other areas of Computer Science including computational complexity, automated theorem proving and learning theory. I have revealed several connections between propositional calculus and other computational models, which have resulted in several new algorithms as well as lower bounds on various algorithmic paradigms. Besides propositional and computational complexity I am also interested in pure algorithmic applications. In particular, I have designed and analyzed several algorithms unrelated to proof complexity and one public key cryptosystem.

Although my interests are purely in Theoretical Computer Science, the techniques involve some deep mathematical tools such as probability theory, convex analysis and commutative algebra. In what follows I briefly describe the set of problems that I addressed during my graduate and postdoctoral studies and some ideas for future research.

1 Propositional Proof Complexity

Propositional Proof Complexity is the point where Mathematical Logic meets Theoretical Computer Science. On the one hand this field studies logical concepts like provability and satisfiability, while on the other hand it investigates the relation between the two complexity classes **NP** and **coNP**. The interest in this field comes from two sources. First, there is the natural curiosity in determining what universal statements possess succinct certificates that may be checked efficiently (i.e., in polynomial time). Moreover, lower bounds for propositional proof systems yield independence results in some weak fragments of bounded arithmetic. Secondly, the machinery and the intuition developed in propositional complexity may have applications in broader areas of computer science, for example in automated theorem proving, or in proving lower bounds in computational complexity.

Lower bounds on the length of proofs. Studying propositional proofs under the supervision of Alexander A. Razborov, I have participated in several research projects for proving lower bounds on the size and space for such systems as Resolution and Polynomial Calculus [17, 16, 15, 13, 12]. Recently I showed that a random 3CNF has no short refutation in Res(k) propositional system [5].

Automatizability of proof systems. The concept of automatizability of proof systems is dedicated to the following basic question: how difficult is it to find a proof provided that one exists. This question is important for *automated theorem proving* because the prover here is a deterministic

algorithm and in practice we are interested in the running time of the prover rather than in the optimal proof length in a given proof system.

In a sequence of works [18, 14, 11] I investigated this direction and obtained several positive and negative results on the existence of automatizing algorithms. In particular with Alexander Razborov we showed that the popular Resolution proof system is not automatizable unless **NP** has subexponential randomized algorithms [14]. In a recent work [6], we adopted these techniques for proving lower bounds in *learning theory*. I elaborate on this result in Section 3.

2 Computational Complexity

Computational Complexity studies the limitations of efficient algorithms to solve various computational tasks. Such limitations indicate that polynomial time algorithms are unlikely to exist for some problems. There is also one notable case in which non-existence of efficient algorithms for certain computational tasks may be useful in practice: the field of cryptography. Below are some directions I have pursued in computational complexity.

Probabilistically Checkable Proofs. The theory of probabilistically checkable proofs (PCP) is a young and active area of research. PCPs allow one to prove formal limitations on the power of approximation algorithms for various NP-complete problems. Recently we showed NP-hardness of approximation within any constant for Nearest Codeword with preprocessing [1]. The latter is an important problem in coding theory that received much attention recently. Our result significantly improves upon the previously known factor 3 on the hardness of approximation.

Weak models of computation. The holy grail of computational complexity is proving that $\mathbf{P} \neq \mathbf{NP}$. In the current state of the art however we are very far from proving such statements, or even proving that **NP** has no superlinear time algorithms. One of the possible explanations of this phenomenon is that we try to prove the statement for *all possible* computer programs; however most programs look like a totally awkward set of instructions. Even though nobody expects a "randomly looking" program to solve an NP-complete problem, this is extremely difficult to prove formally. However existing "meaningful" algorithms usually have some certain inner structure, unlike "random" programs. This suggests that such structure may be used for proving lower bounds on *specific families* of algorithms.

As an example, there are two general algorithmic paradigms that dominate in most of the existing approximation algorithms for NP-complete problems. These are linear or semidefinite relaxation techniques and greedy methods. I have been involved in research projects on understanding the power of the former [4] and the latter [3] algorithmic techniques. We have proved several new tight results on the power of these algorithms for the approximation of NP-complete problems. There still remain many interesting open questions, which look promising for future research.

Cryptography out of a linear mapping. In one of my works [8] I ask about the complexity of a linear mapping in the presence of noise. This paper poses an extended conjecture, stating that a small random perturbation of a linear operator has nice pseudo-random properties. Under this assumption, [8] gives a construction of a simple pseudo-random generator and a public-key cryptosystem. The latter is based on random linear codes and looks similar to the McEllice cryptosystem. While it seems difficult to prove or refute the conjecture for general polynomial time

algorithms, we managed to show that weak local computational models, e.g. DPLL type algorithms, do not break these generators [7].

Parameterized Complexity. Another interesting framework which still seems to contain many exciting open problems is parameterized complexity. Very informally, it studies computational problems that contain a parameter k, which is supposed to be a very slowly growing function. An algorithm solving the problem is considered *tractable* if it runs in time $f(k)n^{O(1)}$, where f is a function that does not depend on n. For example, a typical problem in parameterized complexity may look like this: find a clique of size 20 in a given large graph. The question of existence of efficient parameterized algorithms for a given problem is closely related with the existence of subexponential algorithms for the problem.

In [14] we show the parameterized hardness of approximating the minimum number of ones forcing a given monotone circuit to be true. In [6] we outline connections between parameterized approximation of Set Cover and learning so-called Junta functions. Finally in [2] we prove some partial results on the hardness of approximating Set Cover on small values of the parameter. There is much space for the improvement of this result. One way to tackle the problem would be to design linear size PCP proofs of **NP**.

3 Algorithms

The goal in this research area is the design of new fast algorithms for solving or approximating computational tasks that appear in the real world.

Proper Learning of DNFs and Decision Trees. Computational learning theory is dedicated to the analysis of algorithms for making predictions about the future based on past experiences. A typical task in this field is to approximate an unknown concept f given indirect random access to f w.r.t. probabilistic distribution D. The goal is to produce an *approximator* of f that is consistent with f with high probability.

In [6] we consider the goal of *proper learning*. In this case the hypothesis produced by a learning algorithm should belong to the same concept class as an unknown concept f. For example, given random access to a DNF f one need to produce an approximator $\hat{f}$ which is also a DNF. We present several new upper bounds and show that they are essentially *tight* by proving the corresponding lower bounds for proper learning of DNFs and formulas computed by small decision trees. There are several ways to generalize our results, which may lead to new interesting learning algorithms.

Algorithms for Satisfiability. Satisfiability is one of the most canonical NP complete problems. Given a propositional formula in conjunctive normal form the goal is to find an assignment that satisfies the formula. This problem has been studied by computer scientists, probability theorists and physicists. Of particular interest is the analysis of existing algorithms on a *randomly generated* 3CNF formula. In [9] we showed rigorously that a popular SAT algorithm based on the random walk solves Satisfiability in *linear time* for small density random formulas. Our analysis is based upon geometric reasoning in the space $\mathbf{R}^n$, where n is the number of variables involved.

Another popular algorithmic direction is the design of divide-and-conquer algorithms that exploit the so-called *tree-width* decomposition of the underlying structure. In [11] we show how to

construct an automated theorem prover that runs fast on those boolean formulas, for which the underlying hypergraph has bounded tree-width.

List decoding of Reed-Solomon Codes. List decoding of error correcting codes concerns the reconstructing of a codeword from a corrupted transmitted instance in which the number of errors is too big to hope for an unambiguous decoding. During my graduate education, Madhu Sudan suggested trying to generalize the classical Welch-Berlekamp decoding algorithm for the list decoding of Reed-Solomon codes. I constructed a list-decoding algorithm that runs in quasilinear time [10]. This work (as well as [13]) uses techniques from commutative algebra.

References

[1] Michael Alekhnovich, Subhash Khot, Guy Kindler, Nisheeth Vishnoi. Hardness of approximating the closest vector problem with pre-processing. Manuscript, 2004.

[2] Michael Alekhnovich, Subhash Khot, Toniann Pitassi. Inapproximability of Fixed Parameter Problems. Manuscript, 2004.

[3] Michael Alekhnovich, Allan Borodin, Joshua Buresh-Oppenheim, Russell Impagliazzo, Avner Magen, Toniann Pitassi. Toward a Model for Backtracking and Dynamic Programming. To appear in *Proc. 20th Annual Conference on Computational Complexity*, 2005.

[4] Michael Alekhnovich, Sanjeev Arora, Iannis Tourlakis. Towards strong nonapproximability results in the Lovász-Schrijver hierarchy. To appear in *Proc. 37th Annual ACM STOC*, 2005.

[5] Michael Alekhnovich. Lower bounds for k-DNF Resolution on random 3-CNFs. To appear in *Proc. 37th Annual ACM STOC*, 2005.

[6] Michael Alekhnovich, Mark Braverman, Vitaly Feldman, Adam Klivans, Toniann Pitassi. Learnability and Automatizability. Invited to *JCSS* special issue on learning theory papers. See also *Proc. 44th IEEE Symposium on FOCS*, 2004.

[7] Michael Alekhnovich, Edward A. Hirsch, Dmitry Itsykson. Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. To appear in 2005 SAT special issue of the *Journal of Automated Reasoning*. See also *Proc. 31st ICALP*: 84-96, 2004

[8] Michael Alekhnovich. More on Average Case vs Approximation Complexity. In *Proc. 44th IEEE Symposium on FOCS*: 298-307, 2003.

[9] Michael Alekhnovich, Eli Ben-Sasson. Linear Upper Bounds for Random Walk on Small Density Random 3-CNF. In *Proc. 44th IEEE Symposium on FOCS*: 352-361, 2003.

[10] Michael Alekhnovich. Linear Diophantine Equations over Polynomials and Soft Decoding of Reed-Solomon Codes. To appear in *IEEE Transactions on Information Theory*. See also *Proc. 43rd IEEE Symposium on FOCS*: 439-448, 2002.

[11] Michael Alekhnovich, Alexander A. Razborov. Satisfiability, Branch-Width and Tseitin Tautologies. In *Proc. 43rd IEEE Symposium on FOCS*: 593-603, 2002.

[12] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, Alasdair Urquhart. An exponential separation between regular and general resolution. In *Proc. 34th Annual ACM STOC*: 448-456, 2002.

[13] Michael Alekhnovich, Alexander A. Razborov. Lower Bounds for Polynomial Calculus: Non-Binomial Case. In *Proceedings of the Steklov Institute of Mathematics* 242: 18-35, 2003. See also *Proc. 42nd IEEE Symposium on FOCS*: 190-199, 2001.

[14] Michael Alekhnovich, Alexander A. Razborov. Resolution is Not Automatizable Unless W[P] is Tractable. In *Proc. 42nd IEEE Symposium on FOCS*: 210-219, 2001.

[15] Michael Alekhnovich. Mutilated chessboard problem is exponentially hard for resolution. Michael Alekhnovich. *Theoretical Computer Science* 310(1-3): 513-525, 2004.

[16] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, Avi Wigderson. Pseudorandom Generators in Propositional Proof Complexity. *SIAM Journal on Computing* 34(1): 67-88, 2004. See also *Proc. 41st IEEE Symposium on FOCS*: 43-53, 2000.

[17] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, Avi Wigderson. Space Complexity in Propositional Calculus. *SIAM Journal on Computing* 31(4): 1184-1211, 2002. See also *Proc. 32nd Annual ACM STOC*: 358-367, 2000.

[18] Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, Toniann Pitassi. Minimum Propositional Proof Length is NP-Hard to Linearly Approximate. *Journal of Symbolic Logic* 66(1): 171-191, 2001. See also *Proc. 23rd International Symposium MFCS*: 176-184, 1998.