Minimum Propositional Proof Length is NP-Hard to Linearly Approximate (Extended Abstract)

Michael Alekhnovich[1*], Sam Buss[2**],
Shlomo Moran[3***], and Toniann Pitassi[4†]

[1] Moscow State University, Russia, michael@mail.dnttm.ru
[2] University of California, San Diego, sbuss@ucsd.edu
[3] Technion, Israel Institute of Technology, moran@cs.technion.ac.il
[4] University of Arizona, Tucson, toni@cs.arizona.edu

Abstract. We prove that the problem of determining the minimum propositional proof length is NP-hard to approximate within any constant factor. These results hold for all Frege systems, for all extended Frege systems, for resolution and Horn resolution, and for the sequent calculus and the cut-free sequent calculus. Also, if NP is not in $QP = DTIME(n^{\log^{O(1)} n})$, then it is impossible to approximate minimum propositional proof length within a factor of $2^{\log^{(1-\varepsilon)} n}$ for any $\varepsilon > 0$. All these hardness of approximation results apply to proof length measured either by number of symbols or by number of inferences, for tree-like or dag-like proofs. We introduce the Monotone Minimum (Circuit) Satisfying Assignment problem and prove the same hardness results for Monotone Minimum (Circuit) Satisfying Assignment.

1 Introduction

This paper proves lower bounds on the hardness of finding short propositional proofs of a given tautology and on the hardness of finding short resolution refutations. When considering Frege proof systems, which are textbook-style proof systems for propositional logic, the problem can be stated precisely as the following optimization problem:

Minimum Length Frege Proof:
Instance: A propositional formula φ which is a tautology.
Solution: A Frege proof P of φ.
Objective function: The number of symbols in the proof P.

[*] Supported in part by INTAS grant N96-753

[**] Supported in part by NSF grant DMS-9503247 and grant INT-9600919/ME-103 from NSF and MŠMT (Czech Republic)

[***] Research supported by the Bernard Elkin Chair for Computer Science and by US-Israel grant 95-00238

[†] Supported in part by NSF grant CCR-9457782, US-Israel BSF grant 95-00238, and grant INT-9600919/ME-103 from NSF and MŠMT (Czech Republic)

For a fixed Frege system $\mathcal{F}$, let $\min_{\mathcal{F}}(\varphi)$ denote the minimum number of symbols in an $\mathcal{F}$-proof of φ. An algorithm M is said to approximate the Minimum Length Frege Proof problem within factor α, if for all tautologies φ, $M(\varphi)$ produces a Frege proof of φ of length $\leq \alpha \cdot \min_{\mathcal{F}}(\varphi)$. (Here, α may be a constant or may be a function of the length of φ.)

We are interested only in *polynomial time* algorithms for solving this problem. However, there is a potential pitfall here since the shortest proof of a propositional formula could be substantially longer than the formula itself,[1] and in this situation, an algorithm with runtime bounded by a polynomial of the length of the input could not possibly produce a proof of the formula. In addition, it seems reasonable that a "feasible" algorithm which is searching for a proof of a given length ℓ should be allowed runtime polynomial in ℓ, even if the formula to be proved is substantially shorter than ℓ. Therefore we shall only discuss algorithms that are polynomial time in the length of the shortest proof (or refutation) of the input.

Note that an alternative approach would be to consider a similar problem, **Minimum Length Equivalent Frege Proof**, an instance of which is a Frege proof of some tautology φ, and the corresponding solutions are (preferably shorter) proofs of φ. While our results are all stated in terms of finding a short proof to a given tautology, they hold also for that latter version where the instance is a proof rather than a formula.

A yet different approach could be studying algorithms which output the *size* (i.e., number of symbols) of a short proof of the input formula, rather than the proof itself. In this case it is possible for an algorithm to have run time bounded by a polynomial of the length of the input formula, even if the size of the shortest proof is exponential in the size of the formula. In the final section of this paper, we show that strong non-approximability results can be obtained for algorithms with run time bounded by a polynomial of the length of the formula for a variety of proof systems.

A related minimization problem concerns finding the shortest Frege proof when proof length is measured in terms of the number of steps, or lines, in the proof:

Minimum Step-Length Frege Proof:
Instance: A propositional formula φ which is a tautology.
Solution: A Frege proof P of φ.
Objective function: The number of steps in the proof P.

Resolution is a propositional proof system which is popular as a foundation for automated theorem provers. Since one is interested in finding resolution refutations quickly it is interesting to consider the following problem:

Minimum Length Resolution Refutation
Instance: An unsatisfiable set Γ of clauses.

[1] Is is known that $NP \neq coNP$ implies that some tautologies require superpolynomially long Frege proofs.

Solution: A resolution refutation R of Γ.
Objective function: The number of inferences (steps) in R.

The main results of this paper state that a variety of minimum propositional proof length problems, including the Minimum Length Frege Proof, the Minimum Step-Length Frege Proof and the Minimum Length Resolution Refutation problems, cannot be approximated to within a constant factor by any polynomial time algorithm unless $P = NP$. Furthermore, for these proof systems and for every constant ϵ, the Minimum Length Proof problems cannot be approximated to within a factor of $\epsilon \ln n$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$ or to within a factor of $2^{\log^{(1-\varepsilon)} n}$ unless $NP \subseteq QP$, where QP, quasi-polynomial time, is defined to equal $DTIME(2^{(\log n)^{O(1)}})$. Our results apply to all Frege systems, to all extended Frege systems, to resolution, to Horn clause resolution, to the sequent calculus, and to the cut-free sequent calculus; in addition, they apply whether proofs are measured in terms of symbols or in terms of steps (inferences), and they apply to either dag-like or tree-like versions of all these systems.

We let $\mathcal{F} \overset{k}{\vdash} \varphi$ mean that φ has an $\mathcal{F}$-proof of $\leq k$ symbols. One of the first prior results about the hardness of finding optimal length of Frege proofs was the second author's result [7] that, for a particular choice of Frege system $\mathcal{F}_1$ with the language $\wedge$, $\vee$, $\neg$ and $\rightarrow$, there is no polynomial time algorithm which, on input a tautology φ and a $k > 0$, can decide whether $\mathcal{F}_1 \overset{k}{\vdash} \varphi$, unless P equals NP. This result however applies only to a particular Frege system, and not to general Frege systems. It also did not imply the hardness of approximating Minimum Length Frege Proofs to within a constant factor.

A second related result, which follows from the results of Krajíček and Pudlák [13], is that if the RSA cryptographic protocol is secure, then there is no polynomial time algorithm for approximating the Minimum Step-Length Frege Proof problem to within a polynomial.

Another closely related prior result is the striking connection between the (non)automatizability of Frege systems and the (non)feasibility of factoring integers that was recently discovered by Bonet-Pitassi-Raz [6]. A proof system T is said be *automatizable* provided there is an algorithm M and a polynomial p such that whenever $T \overset{n}{\vdash} \varphi$ holds, $M(\varphi)$ produces some T-proof of φ in time $p(n)$ (see [8]). Obviously the automatizability of Frege systems is closely related to the solution of the Minimum Length Frege Proof problem. Our theorems give a linear or quasi-linear lower bound on the automatizability of the Minimum Proof Length problem based on the assumption that $P \neq NP$ or that $NP \not\subseteq QP$. It has recently been shown by Bonet-Pitassi-Raz [6] that Frege systems are not automatizable unless Integer Factorization is in P. Their result provides a stronger non-approximability conclusion, but requires assuming a much stronger complexity assumption.

For resolution, the first prior hardness result was Iwama-Miyano's proof in [11] that it is NP-hard to determine whether a set of clauses has a read-once refutation (which is necessarily of linear length). Subsequently, Iwama [10] proved that it is in NP-hard to find shortest resolution refutations; unlike us, he did not obtain an approximation ratio bounded away from 1.

2 Monotone Minimum Satisfying Assignment

The section introduces the Monotone Minimum Satisfying Assignment problem and shows it is harder to approximate than the Minimum Set Cover problem and the Minimum Label Cover. (The latter is needed for proving hardness of approximation within a superlinear factor). The reader can find a general introduction to and survey of the hardness of approximation and of probabilistically checkable proofs in [4] and [2]. Recall that an A-reduction, as defined by [12], is a polynomial-time Karp-reduction which preserves the non-approximating ratio to within a constant factor.

Consider the following *NP*-optimization problems:

Monotone Minimum Satisfying Assignment:
Instance: A monotone formula $\varphi(x_1, \ldots, x_n)$ over the basis $\{\vee, \wedge\}$.
Solution: An assignment $\langle v_1, \ldots, v_n \rangle$ such that $\varphi(v_1, \ldots, v_n) = \top$.
Objective function: The number of v_i's which equal $\top$.

We henceforth let $\rho(\varphi)$ denote the value of the optimal solution for the Monotone Minimum Satisfying Assignment problem for φ i.e., the minimum number of variables v_i which must be set *True* to force φ to have value *True*.

We will also consider the Monotone Minimum Circuit Satisfying Assignment problem which is to find the minimum number of variables which must be set *True* to force a given monotone circuit over the basis $\{\wedge, \vee\}$ evaluate *True*. It does not matter whether we consider circuits with bounded fanin or unbounded fanin since they can simulate each other. It is apparent that Monotone Minimum Circuit Satisfying Assignment is at least as hard as Monotone Minimum Satisfying Assignment.

Recall the Minimum Hitting Set problem, which is:

Minimum Hitting Set:
Instance: A finite collection $\mathcal{S}$ of nonempty subsets of a finite set U.
Solution: A subset V of U that intersects every member of $\mathcal{S}$.
Objective function: The cardinality of V.

It is easy to see that Monotone Minimum Satisfying Assignment is at least as hard as Minimum Hitting Set: namely Minimum Hitting Set can be reduced (via an A-reduction) to the special case of Monotone Minimum Satisfying Assignment where the propositional formula is in conjunctive normal form. Namely, given $\mathcal{S}$ and U, identify members of U with propositional variables and form a CNF formula which has, for each set in $\mathcal{S}$, a conjunct containing exactly the members of that set.

Lund and Yannakakis [14] noted that Minimum Hitting Set is equivalent to Minimum Set Cover (under A-reductions). Furthermore, it is known that the problem of approximating Minimum Set Cover to within any constant factor is not in polynomial time unless $P = NP$ [5]. If one makes a stronger complexity assumption, then one can obtain a better non-approximability result for Minimum Set Cover; namely, Feige [9] has proved that Minimum Set Cover cannot be approximated to within a factor of $(1 - \epsilon) \ln n$ unless $NP \subseteq DTIME(n^{O(\log \log n)})$.

In fact, we can get stronger results than the above reduction of Minimum Set Cover to Monotone Minimum Satisfying Assignment. There are two ways to see this: firstly, we can use a construction due to S. Arora [private communication] to reduce Monotone Minimum Satisfying Assignment to the Minimum Label Cover problem, or alternatively we can use a "self-improvement" property of the Monotone Minimum Satisfying Assignment problem to directly prove better non-approximation results. Both approaches prove that Monotone Minimum Satisfying Assignment cannot be approximated to within a factor of $2^{(\log n)^{1-\epsilon}}$ unless $NP \subseteq QP$. The advantage of the first approach is that it gives a sharper result, namely, a reduction of Minimum Label Cover to Monotone Minimum Satisfying Assignment for Π_4-formula. The second approach is more direct in that it avoids the use of Label Cover. (We include details of the second approach in the full version of this paper, but not in this abstract.)

Minimum Label Cover: (see [2])

Instance: The input consists of: (i) a regular bipartite graph $G = (U, V, E)$, (ii) an integer N in unary, and (iii) for each edge $e \in E$, a partial function $\Pi_e : \{1, \ldots, N\} \rightarrow \{1, \ldots, N\}$ such that 1 is in the range of Π_e.

The integers in $\{1, \ldots, N\}$ are called *labels*. A *labeling* associates a nonempty set of labels with every vertex in U and V. A labeling *covers* an edge $e = (u, v)$ (where $u \in U$, $v \in V$) iff for every label ℓ assigned to v, there is some label t assigned to u such that $\Pi_e(t) = \ell$.

Solution: A labeling which covers all edges.

Objective function: The number of all labels assigned to vertices in U and V.

A *Π_4-formula* is a propositional formula which is written as an AND of OR's of AND's of OR's.

Theorem 1 (S. Arora) *There is an A-reduction from Minimum Label Cover to Monotone Minimum Satisfying Assignment such that the instances of Label Cover are mapped to Π_4 formulas.*

For space reasons, we omit the proof of this theorem.

It was proved in [1] that Minimum Label Cover is not approximable within a $2^{\log^{(1-\varepsilon)} n}$ factor unless $NP \subseteq QP$. An immediate corollary of Theorem 1 is that Monotone Minimum Satisfying Assignment enjoys the same hardness of approximation, even when restricted to Π_4-formulas. Summarizing, we have

Theorem 2

(a) *If $P \neq NP$, then there is no polynomial time algorithm which can approximate Monotone Minimum Satisfying Assignment (and hence Monotone Minimum Circuit Satisfying Assignment) to within a constant factor.*

(b) *If $NP \not\subseteq DTIME(n^{O(\log \log n)})$, then Monotone Minimum Satisfying Assignment (and Monotone Minimum Circuit Satisfying Assignment) cannot be approximated to within a factor of $(1-\epsilon)\ln n$ where n equals the number of distinct variables.*

(c) *If $NP \not\subseteq QP$, then there is no polynomial time algorithm which can approximate Monotone Minimum Satisfying Assignment (or Monotone Minimum Circuit Satisfying Assignment) to within a factor of $2^{\log^{(1-\varepsilon)} n}$.*

The main theorems of this paper are stated in the next section. Their proofs depend on the reduction of the Monotone Minimum Circuit Satisfying Assignment problem to problems on minimum T-proof length, for a variety of propositional proof systems T.

Open question. Is it possible to improve the non-approximation factor for Monotone Minimum Satisfying Assignment or Monotone Minimum Circuit Satisfying Assignment, or prove their hardness using just $P \neq NP$ as a complexity theory hypothesis?

In fact the known NP-hardness of Monotone Minimum Satisfying Assignment concerns Π_2 (CNF) formulae and Quasi NP-hardness uses Π_4 formulae. But in general the formula or circuit can have unbounded depth and thus it a priori has richer expressive abilities. Hence there could be some chance to prove its hardness by some other way without improving the corresponding factor of Label Cover, perhaps by using some extension of the self-improvement property.

3 Main Hardness Results

Our first main results state that it is hard to approximate the length of the shortest T-proof of a given tautology in a wide variety of propositional proof systems T.

Hardness Theorem 3 *Let T be one of the following propositional proof systems: (1) a Frege system, (2) an extended Frege system, (3) resolution, (4) Horn clause resolution, (5) the sequent calculus, or (6) the cut-free sequent calculus. Let T-proofs have length measured by either (a) number of symbols, or (b) number of steps (lines). Finally, for each system, we may either require proofs to be tree-like or allow them to be dag-like. (So overall, there are 24 possible choices for the system T.)*

(a) *If $P \neq NP$, then there is no polynomial time algorithm which can approximate Minimum Length T-Proof to within a constant factor.*

(b) *If $NP \not\subseteq DTIME(n^{O(\log \log n)})$, then there is a $c > 0$ such that there is no polynomial time algorithm which can approximate Minimum Length TProof to within a factor of $c \cdot \log n$.*

(c) *If $NP \not\subseteq QP$, then there is no polynomial time algorithm which can approximate Minimum Length TProof to within a factor of $2^{\log^{(1-\varepsilon)} n}$ for any ε.*

The proof of the Hardness Theorem 3 involves giving a reduction of the Monotone Minimum (Circuit) Satisfying Assignment problem to the Minimum Length T-Proof problem. Thus any hardness results for the Monotone Minimum Satisfying Assignment or Monotone Minimum Circuit Satisfying Assignment problem immediately also apply to the Minimum Length Frege proof problem.

For space reasons, the proofs are omitted from this abstact, but they are already available in the full version of the paper.

4 Hardness results for long proofs

In the previous sections we proved that it is NP-hard to approximate the minimal propositional proof length by any constant factor, and that if NP is not in QP then minimum proof-length cannot be approximated (in polynomial time) within a $2^{\log^{(1-\varepsilon)} n}$ factor. The tautologies used in the proofs of these results had "short" proofs (or refutations); that is, proofs whose length is polynomial in the size of the formula. However, if $NP \neq coNP$, then for any proof system $\mathcal{S}$, there are tautologies whose shortest $\mathcal{S}$-proof is of super-polynomial length. It is therefore interesting whether better non-approximability results can be achieved when the proof lengths are not bounded, and when the run time of the algorithm is required to be polynomial time in the length of the input formula only.

The following simple intuition implies that in this case, no polynomial time algorithm can guarantee a polynomial time approximation for the shortest refutation of a given unsatisfiable formula, unless $NP \not\subseteq P/poly$: [2]

Given an input formula ψ of length n, reduce it to a formula $\varphi = \psi \wedge \eta$, such that the size of η is polynomial in that of ψ, η is unsatisfiable, but its shortest refutation is longer than the shortest refutation of any unsatisfiable formula of length n by a super-polynomial factor. Then ψ is satisfiable iff on input φ, a supposed polynomially bounded approximation algorithm returns a number smaller than than the size of the shortest refutation of φ. This implies a polynomial time circuit for recognizing SAT. To make the above argument formal, we need few more definitions.

Definition 1. *For a proof system $\mathcal{S}$ and an unsatisfiable formula φ, $\min_{\mathcal{S}}(\varphi)$ is the minimum length of a refutation of φ in $\mathcal{S}$. For an integer n, $MAX_{\mathcal{S}}(n) = \max\{\min_{\mathcal{S}}(\varphi)\}$, where φ ranges over all unsatisfiable formulas of length $\leq n$.*

We say that a non-decreasing function f has *super-polynomial growth* if for every polynomial r, $f(n) > r(n)$ for almost all positive integers n. f has a *smooth* super-polynomial growth if in addition there is a constant D such that for each large enough n there is $1 < d < D$ such that $f(n^d) > f^d(n)$. [If we write $f(n) = n^{e(n)}$, then the first condition states that $e(n)$ is not bounded from

[2] We present the results in terms of finding short refutations of unsatisfiable formulas, but equivalent definitions and results are easily obtained for finding short proofs of tautologies.

above, and the second condition states that for each n there is m, $n < m < n^D$, such that $e(m) > e(n)$.]

Assume, for simplicity, that $\mathcal{S}$ contains the connective $\wedge$. Formulas ψ and η are said to be *disjoint* if their underlying sets of variables are disjoint.

Theorem 4 *Assume that $NP \not\subseteq P/poly$, and let $\mathcal{S}$ be a proof system which satisfies:*

1. *For every pair of disjoint formulas ψ and η, where η is unsatisfiable, the following holds:*
   (a) If ψ is unsatisfiable, then $\min_{\mathcal{S}}(\psi \wedge \eta) < \min_{\mathcal{S}}(\psi) + r(|\psi| + |\eta|)$ for some (fixed) polynomial r.
   (b) If ψ is satisfiable, than $\min_{\mathcal{S}}(\psi \wedge \eta) \geq \min_{\mathcal{S}}(\eta)$;
2. *$MAX_{\mathcal{S}}(n)$ has a smooth super-polynomial growth.*

Then for any polynomial q, there is no polynomial time q-approximation algorithm for the minimum length proof in $\mathcal{S}$.

Observe that property 1 above holds trivially for all proof systems mentioned in this paper. Property 2 is known to hold for resolution, since in this case $MAX_{\mathcal{S}}(n) < 3^n$ for all n, and by [3], for each n there is an e, $1 < e < 3$, s.t. $MAX_{\mathcal{S}}(n^e) > 2^{\frac{n^e}{40}}$, thus property 2 holds for $D = 3$. We conjecture that this property holds for any known proof system in which the proof lengths are not polynomially bounded.

Proof. We show that the existence of a polynomial time q-approximation algorithm, AL, for $\mathcal{S}$, implies polynomial size circuits for solving SAT.

Let j be such that $q(n) < n^j$ for almost all n, and let D be the constant guaranteed by the smooth super-polynomial growth of $MAX_{\mathcal{S}}$. Since $MAX_{\mathcal{S}}$ has super-polynomial growth, for all large enough n it holds that $r(n + n^{2jD}) < MAX_{\mathcal{S}}(n)$. Fix an integer n_0 for which this inequality holds. Since the super-polynomial growth of $MAX_{\mathcal{S}}$ is smooth, there is a number d, $2j \leq d \leq 2jD$, such that $[MAX_{\mathcal{S}}(n_0)]^d < MAX_{\mathcal{S}}(m)$, where $m = {n_0}^d$. Let η_m be a formula of size $\leq m$ such that $\min_{\mathcal{S}}(\eta_m) = MAX_{\mathcal{S}}(m)$. An input formula ψ of size n_0 is reduced to $\varphi = \psi \wedge \eta_m$, where the variables of η_m are disjoint from these of ψ (note that φ is unsatisfiable and its size is polynomial in that of ψ). We claim that ψ is unsatisfiable if and only if AL on input φ will output a number $k < MAX_{\mathcal{S}}(m)$. To see this, observe that if ψ is unsatisfiable, then by property (1a) above, $\min_{\mathcal{S}}(\varphi) \leq \min_{\mathcal{S}}(\psi) + r(|\psi| + |\eta_m|) < 2MAX_{\mathcal{S}}(n_0)$. Hence, by the assumption on AL, AL must produce an output $k < (2MAX_{\mathcal{S}}(n_0))^j < MAX_{\mathcal{S}}(m) = \min_{\mathcal{S}}(\eta_m)$. On the other hand, if ψ is satisfiable, then, by property (1b), $\min_{\mathcal{S}}(\varphi) \geq \min_{\mathcal{S}}(\eta_m) = MAX_{\mathcal{S}}(m)$.

5 Acknowledgments

We are grateful to A.A. Razborov for extremely helpful discussions. We also would like to thank S. Arora for pointing out that Minimum Label Cover can be reduced to Monotone Minimum Satisfying Assignment.

References

1. S. Arora, L. Babai, J. Stern, and Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, Journal of Computer and System Sciences, 54 (1997), pp. 317–331. Earlier version in *Proc. 34th Symp. Found. of Comp. Sci.*, 1993, pp.724-733.
2. S. Arora and C. Lund, *Hardness of approximations*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS Publishing Co., Boston, 1996, p. ???
3. P. Beame and T. Pitassi, *Simplified and improved resolution lower bounds*, in Proceedings, 37th Annual Symposium on Foundations of Computer Science, Los Alamitos, California, 1996, IEEE Computer Society, pp. 274–282.
4. M. Bellare, *Proof checking and approximation: Towards tight results*, SIGACT News, 27 (1996), pp. 2–13. Revised version at `http://www-cse.ucsd.edu/users/mihir`.
5. M. Bellare, S. Goldwasser, C. Lund, and A. Russell, *Efficient probabalistically checkable proofs and applications to approximation*, in Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, 1993, pp. 294–304.
6. M. L. Bonet, T. Pitassi, and R. Raz, *No feasible interpolation for TC^0-Frege proofs*, in Proceedings of the 38th Annual Symposium on Foundations of Computer Science, Piscataway, New Jersey, 1997, IEEE Computer Society, pp. 264–263.
7. S. R. Buss, *On Gödel's theorems on lengths of proofs II: Lower bounds for recognizing k symbol provability*, in Feasible Mathematics II, P. Clote and J. Remmel, eds., Birkhäauser-Boston, 1995, pp. 57–90.
8. M. Clegg, J. Edmonds, and R. Impagliazzo, *Using the Groebner basis algorithm to find proofs of unsatisfiability*, in Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing, Association for Computing Machinery, 1996, pp. 174–183.
9. U. Feige, *A threshold of $\ln n$ for approximating set cover*, in Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, 1996, pp. 314–318.
10. K. Iwama, *Complexity of finding short resolution proofs*, in Mathematical Foundations of Computer Science 1997, I. Prívara and P. Ruzicka, eds., Lecture Notes in Computer Science #1295, Springer-Verlag, 1997, pp. 309–318.
11. K. Iwama and E. Miyano, *Intractibility of read-once resolution*, in Proceedings of the Tenth Annual Conference on Structure in Complexity Theory, Los Alamitos, California, 1995, IEEE Computer Society, pp. 29–36.
12. S. Khanna, M. Sudan, and L. Trevisan, *Constraint satisfaction: The approximability of minimization problems*, in Twelfth Annual Conference on Computational Complexity, IEEE Computer Society, 1997, pp. 282–296.
13. J. Krajíček and P. Pudlák, *Some consequences of cryptographic conjectures for S_2^1 and EF*, in Logic and Computational Complexity, D. Leivant, ed., Berlin, 1995, Springer-Verlag, pp. 210–220.
14. C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, Journal of the Association for Computing Machinery, 41 (1994), pp. 960–981.