More on average case vs approximation complexity

Michael Alekhnovich *

April 13, 2003

Abstract

We consider the problem to determine the maximal number of satisfiable equations in a linear system chosen at random. We make several plausible conjectures about the average case hardness of this problem for some natural distributions on the instances, and relate them to several interesting questions in the theory of approximation algorithms and in cryptography. Namely we show that our conjectures imply the following facts:

- Feige's hypothesis about the hardness of refuting a random 3CNF is true, which in turn implies inapproximability within a constant for several combinatorial problems, for which no **NP**-hardness of approximation is known.
- It is hard to approximate the NEAREST CODEWORD within factor $n^{1-\epsilon}$.
- It is hard to estimate the rigidity of a matrix. More exactly, it is hard to distinguish between matrices of low rigidity and random ones.
- There exists a secure public-key (probabilistic) cryptosystem, based on the intractability of decoding of random binary codes.

Our conjectures are strong in that they assume cryptographic hardness: no polynomial algorithm can solve the problem on any non-negligible fraction of inputs. Nevertheless, to the best of our knowledge no efficient algorithms are currently known that refute any of our hardness conjectures.

1. Introduction

Since the discovery of the PCP theorem in the beginning of 90s ([AS92],[ALMSS98]), there has been much progress in proving the hardness of approximating optima for various classes of combinatorial problems. This research lead to many brilliant results, for many problems the optimal hardness of approximation that matches the upper bounds given by approximation algorithms has been achieved. However in some cases there is still a large gap between known upper and lower bounds. In some cases (see [Has88], [LLS90], [GG98]) it is unlikely to show NP-hardness of approximation within

*Laboratory for Computer Science, MIT, mishka@theory.lcs.mit.edu. Supported in part by NSF Awards CCR 0205390 and MIT NTT Award 2001-04.

factor beyond a certain barrier. Thus, it is a natural goal to investigate the hardness of such optimization problems in some other frameworks different from **NP**-completeness.

Recently, Feige [Fei02] suggested to use cryptographic conjectures for proving interesting inapproximability results. His method is based on the observation that if an instance of certain NP-complete problems looks "like random" then more approximation preserving reductions can be constructed, that donot work on the arbitrary instance. It was assumed in [Fei02] that it is hard to refute in polynomial time a random 3CNF with linearly many clauses. On one hand this hypothesis is much stronger than the usual worst case hardness assumptions. In particular, by itself it immediately implies the hardness of approximating MAX-3SAT within the optimal constant $8/7-\epsilon$. On the other hand, this hypothesis implies inapproximability for problems, for which no NP-hardness of approximation is known, which makes it a promising and interesting direction for the further investigation of tractability of NP-complete problems.

In this paper we continue the research initiated by Feige and show more relations between the average case complexity and the complexity of approximation. In order to investigate this direction further it is convenient to define a uniform framework that would embrace both the average case and the worst case complexity. For this we suggest to use the notion of a promise problem generalized for the probabilistic case. In this new concept the instance of the problem is "promised" to be chosen according to one of random distributions, that belong to some given family. The most important example of such a problem is the classical cryptographic task to distinguish two distributions with non-negligible success probability.

In the base of our considerations lies the problem of maximizing the number of satisfied equations in a linear system. Due to its high symmetry over GF_2 field the linear mapping possesses some nice "pseudorandom" and pointwise independent properties. The powerful gaussian elimination procedure can invert a linear mapping, however if one adds small non-linear noise to the system, the resulting function becomes hard to invert or to decode in general (one exception is efficiently decodable linear error correcting codes, however only few codes have known polynomial decoding algorithms). The complexity of a linear mapping (sometimes augmented with small number of "non-linear" errors) was considered by many researchers and successfully used in several theoretical and practical applications. To name just a few, this includes

- Hastad's PCP [Has01]. His construction in particular shows that it is NP-hard to approximate MAX-3LIN within the optimal factor $1/2-\epsilon$. This implies that nothing better than a random guessing is possible to maximize the number of satisfiable linear constraints over 3 variables.
- Classical construction of (almost) pointwise independent families using linear codes, see for example [NN93].
- Tseitin tautologies for propositional calculus. Since the seminal paper [Tse68] there has been proved a lot of lower bounds on refuting an unsatisfiable linear system for many propositional proof systems.
- Property testing [BHR03]. This recent work provides examples of 3CNF properties based on linear functions that are hard to test.
- Practical cryptographic applications: linear feedback shift registers. In these practical constructions an output of finite automata computing a linear opera-

tor augmented by some nonlinear transform is used as a very fast generator of pseudorandom bits, see for example [GC89].

In this paper we consider the following problem: distinguish vectors located within hamming distance k from the linear space $\mathrm{Im}(A)$ from those located within distance $k+1$ from $\mathrm{Im}(A)$, where $A : \{0,1\}^n \to \{0,1\}^m$ is a linear operator over GF_2. We make several conjectures about the hardness of this problem on average for various types of A. As a corollary we derive the following hardness results.

First, if the problem to compute the distance to $\mathrm{Im}(A)$ is hard on average for a random sparse matrix A (which doesnot contradict to the current state of the art in efficient algorithms) then it is hard to refute a random 3CNF on average, i.e. Feige's assumption is true. We believe that this result brings more evidence for Feige's hypothesis.

Under a similar (strong) conjecture we prove that it is hard to approximate the NEAREST CODEWORD problem within factor $n^{1-\epsilon}$. As a consequence of this result we show the hardness of estimating the rigidity of a given matrix. Since this is one of our main motivations we would like to elaborate more on this concept.

As defined by Valiant [Val77], the rigidity $\mathcal{R}_M(r)$ of $(0-1)$–matrix M is the minimal number of entries of M that have to be changed to reduce its rank below r. This notion is tightly connected to the linear circuit complexity, namely Valiant showed that for any sequence of matrices M_n s.t. $\mathcal{R}_{M_n}(\epsilon n) > n^{1+\delta}$, the vector multiplication by M_n cannot be performed by linear circuits of linear size and logarithmic depth. Since then a lot of research has been done towards the understanding the notion of rigidity ([Raz89], [PV91], [Fri93], [SSS97], [KR98], [Lok01]), however still there are no known explicit constructions of matrices with high rigidity. We try to explain the intricate difficulty to find such explicit matrices from the point of the natural proofs approach by Razborov and Rudich [RR97]. We show that it is not likely to prove lower bounds for the rigidity by constructing an efficiently computable property $\mu(M)$ which separates matrices of low rigidity from random ones. We hope that this result may give some evidence that new ("non-natural") ideas are necessary for proving lower bounds on matrix rigidity.

Finally, modulo stronger hypothesis that it is not feasible to decode $n^{1/2-\epsilon}$ errors for a random linear error correcting code we construct two public key cryptosystems, based on binary codes. The first cryptosystem was inspired by Ajtai-Dwork lattice cryptosystem ([AD97]), which gives a brilliant reduction from the worst-case to the average case complexity. Our second cryptosystem is similar to McElice cryptosystem ([M78]), however in our case the choice of the underlying error correcting code is arbitrary and the security is based solely on the assumed hardness of decoding a *random* code. We donot claim (although donot exclude either) any practical significance of the constructed cryptosystems, however we hope that they might be interesting from the theoretical point of view, and probably might have some applications in complexity theory.

The paper is organized in the following way. Section 2 contains some basic notation and the definition of probabilistic promise problems. Section 3 proves our main reduction, which we use for applications in Section 4. We finish our paper with discussion and open questions in Section 5.

2. Preliminaries

We will mainly work in the field GF_2. For $0-1$ vector x, its *hamming weight* is the number of ones in x. We use greek letters for random variables and capital letters for distributions. For a distribution D we write $\xi \sim D$ to indicate that random variable ξ is chosen according to D. We denote the uniform distribution on the set $\{0,1\}^n$ by U_n.

Definition 2.1 (statistical distance) *For two random variables ξ^1 and ξ^2 their statistical distance is defined as*

$$\rho(\xi^1,\xi^2) = \max_{A(x)} \left|\Pr[A(\xi^1)=1] - \Pr[A(\xi^2)=1]\right|,$$

where $A(x)$ is an arbitrary statistical test.

Definition 2.2 (computational distance) *Two sequences of random variables ξ^1_n and ξ^2_n are said to be* computationally $f(n)$-close *iff for any constant k and for any (randomized) algorithm C running in time n^k there exists N s.t. for all $n > N$*

$$|\Pr[C(\xi^1_n)=1] - \Pr[C(\xi^2_n)=1]| \le f(n).$$

Sometimes when clear from the context we omit the lower index n in the asymptotic notation of random sequences (as well as other objects parameterized by the length of the input). We write $\rho_c(\xi^1,\xi^2) < f(n)$ to indicate that ξ^1 and ξ^2 are computationally $f(n)$-close. Distributions are computationally $f(n)$-close if so are the corresponding random variables. Distributions are *computationally indistinguishable* iff their computational distance is less than $1/n^{\Omega(1)}$.

It is well known that statistical and computational distance are metrics on the space of random variables. In particular, the following fact holds.

Proposition 2.1 *Assume that ξ^1,ξ^2,ξ^3 are random sequences satisfying $\rho_c(\xi^1,\xi^2) < f_1(n)$ and $\rho_c(\xi^2,\xi^3) < f_2(n)$. Then*

$$\rho_c(\xi^1,\xi^3) < f_1(n) + f_2(n).$$

A linear binary error correcting code $\mathcal{C}$ is a linear subspace of GF_2^m. As any linear space it can be specified as an image of a linear operator: $\mathcal{C} = \mathrm{Im}(G)$ (in this case G is called the *generator* of the code $\mathcal{C}$) or as a kernel of a linear operator: $\mathcal{C} = \mathrm{Ker}(H)$ (in this case H is called the *parity check matrix* for $\mathcal{C}$). For the code $\mathcal{C}$ generated by G its dual code $\mathcal{C}^\perp$ is the code with parity check matrix G^T.

2.1. Probabilistic promise problems

Promise problems are a useful formalism for proving gaps in approximation of **NP**-hard problems. In a promise problem the instance is "promised" to be taken out of specific subclass of all instances. It is convenient to use the following generalized definition of this notion for the statement of our results.

Definition 2.3 (probabilistic promise problem) *Let Ω be a probability space, Σ a finite alphabet, and Σ^n be the set of all words of length n over Σ. A* probabilistic promise problem *is a sequence* $(\Pi_n^{yes}, \Pi_n^{no})$,

$$\Pi_n^{yes}, \Pi_n^{no} \subseteq (\Sigma^n)^{\Omega},$$

such that for every n, Π_n^{yes} and Π_n^{no} contain only measurable functions $\xi_n : \Omega \to \Sigma^n$.

For an algorithm A and the probabilistic promise problem $(\Pi_n^{yes}, \Pi_n^{no})$ define its completeness *as*

$$c_n = \min_{\xi \in \Pi_n^{yes}} \Pr_{\Omega}[A(\xi) = 1]$$

and its soundness *as*

$$s_n = \max_{\xi \in \Pi_n^{no}} \Pr_{\Omega}[A(\xi) = 1)].$$

The algorithm solves the problem with success probability $f(n)$ *iff for every n $c_n - s_n > f(n)$.*

One can imagine this definition in the following way. Assume that the instance of the promise problem is generated according to some probabilistic distribution, which belongs to some general family (e.g. normal distributions). Then the success of an algorithm is defined as the worst among all distributions in the family. The next example shows that this definition is indeed a generalization of usual promise problems.

Example 1
Given a pair of non-intersecting languages L^{yes} and L^{no} that describe the standard promise problem choose $\Omega = \{0\}$ and let Π_n^{yes} contain the functions $\xi : 0 \mapsto x_n$ for all $x_n \in \Sigma^n \cap L^{yes}$ and Π_n^{no} contain the functions $\xi : 0 \mapsto y_n$ for all $y_n \in \Sigma^n \cap L^{no}$. There is no randomness, the admissible distributions coincide with *yes* and *no* instances and c_n, s_n are always either 0 or 1.

Another example is the problem considered in [Fei02].

Example 2 (refuting a random 3CNF)
Let Π_n^{no} contain only one distribution that chooses a random 3CNF with n variables and Δn clauses, where Δ is a large constant. Π_n^{yes} consists of all distributions for which the generated 3CNF is always satisfiable.

Clearly every algorithm that solves this probabilistic promise problem with high success probability should always say "yes" on every satisfiable CNF and say "no" with high probability on a random CNF.

A probabilistic promise problem is *samplable* iff both Π^{yes} and Π^{no} contain only one distribution which is samplable in polynomial time. It is easy to show that in this case the probabilistic promise problem is equivalent to the standard cryptographic task to distinguish *yes* and *no* distributions:

Proposition 2.2 *Assume that $\Pi = (\Pi^{yes}, \Pi^{no})$, $\Pi_n^{yes} = \{\xi_n^{yes}\}, \Pi_n^{no} = \{\xi_n^{no}\}$ is a samplable promise problem and*

$$\rho_c(\xi^{yes}, \xi^{no}) > f(n),$$

where $f(n) = 1/n^{O(1)}$. Then there exists an algorithm that solves the problem Π with success $f(n)/2$.

Proof. By the statement of the proposition there exists an algorithm A that distinguishes ξ^{yes} and ξ^{no}. The only difficulty is that given A it is not clear whether $\Pr[A(\xi^{yes}) = 1] - \Pr[A(\xi^{no}) = 1] > f(n)$ or $\Pr[A(\xi^{no}) = 1] - \Pr[A(\xi^{yes}) = 1] > f(n)$. However since ξ^{yes}, ξ^{no} are samplable, this can be determined in polynomial time with probability $1 - o(1)$.■

3. Main reduction

In the core of our reductions lies the following **NP**-optimization problem called MAXIMUM SATISFYING LINEAR SUBSYSTEM (MAX-LIN-SAT for short).

Problem 1 (MAXIMUM SATISFYING LINEAR SUBSYSTEM)

- INSTANCE: System $Ax = b$ of linear equations, where A is $m \times n$ matrix over GF_2, and b is a vector in $\{0,1\}^m$.
- SOLUTION: A vector $x \in \{0,1\}^n$.
- OBJECTIVE FUNCTION: The number of equations satisfied by x.

Below we define an average case version of this problem, in which the random system is generated by choosing a planted solution and adding a number of errors.

Definition 3.1 *Let A be $m \times n$ matrix over GF_2. Let $\mathrm{D}_k(A)$ be the distribution of the random vector*

$$\eta_k(A) = Ax + e,$$

where $x \sim U_n$ is random and $e \in \binom{m}{k}$ is randomly chosen from the vectors of hamming weight k.

Thus, $\mathrm{D}_k(A)$ is the distribution of a random vector located within distance k from $\mathrm{Im}(A)$. We are interested in the complexity of maximizing the number of satisfied equations in the system

$$Ax = \eta.$$

More exactly, we want to distinguish between distributions $D_k(A)$ and $D_{k+1}(A)$. Below we show that if this problem is hard then the distribution $D_k(A)$ is computationally close to the uniform, i.e. $\eta_k(A)$ is a good pseudorandom generator.

Theorem 3.1 *Let $k = k(n), m = m(n)$ be integer parameters, $\epsilon = \epsilon(n)$ be a positive real and A_n be a sequence of $m \times n$ matrices over GF_2. Assume that*

$$\rho_c(\mathrm{D}_k(A), \mathrm{D}_{k-1}(A)) < \epsilon \text{ and } \rho_c(\mathrm{D}_k(A), \mathrm{D}_{k+1}(A)) < \epsilon.$$

Then the distribution $\mathrm{D}_k(A)$ is computationally $O(t\epsilon + me^{\Omega(-t/m)})$-close to the uniform for any choice of t.

Proof. We need the following classical example of rapidly mixing random walk:

Definition 3.2 (lazy random walk on the cube) *Let $\xi_1 \in \{0,1\}^m$ be a random vector equal 0 with probability 1/2 and a randomly chosen vector of weight 1 with probability 1/2. This is the "step" of the random walk. Let*

$$\xi_k = \sum_{i=1}^{k} \xi_1^{(i)},$$

where $\xi_1^{(i)} \sim \xi_1$ are i.i.d. variables and the sum is taken over GF_2.

It is well known (see for example [PV01]) that the distribution of ξ_t converges to the uniform very fast, namely the statistical distance

$$\rho(\xi_t, U_m) < me^{-\Omega(t/m)}.$$

Lemma 3.1 *Let $\eta_r(A) \sim D_r(A)$. In the assumption that $\rho_c(\eta_k(A), \eta_{k-1}(A)) < \epsilon$ and $\rho_c(\eta_k(A), \eta_{k+1}(A)) < \epsilon$ holds*

$$\rho_c(\eta_k(A), \eta_k(A) + \xi_1) < \epsilon/2.$$

Proof. Let C be a randomized polynomial algorithm. The variable $\eta_k(A) + \xi_1$ results from $\eta_k(A)$ by flipping a random bit with probability 1/2. Recall that $\eta_k(A) = Ax + e_k$, thus

$$\eta_k(A) + \xi_1 = Ax + (e_k + \xi_1).$$

With probability 1/2 the variable $e_k + \xi_1$ is a uniform vector of weight k, with probability $1/2 \cdot k/n$ it is a uniform vector of weight $k-1$ and with probability $1/2 \cdot (n-k)/n$ it is of weight $k+1$. Denote by $p_r = \Pr[C(\eta_r) = 1]$. Thus, we can write

$$\Pr[C(\eta_k + \xi_1) = 1] = \frac{1}{2}p_k + \frac{k}{2n}p_{k-1} + \frac{n-k}{2n}p_{k+1}$$

which implies that

$$\Big|\Pr[C(\eta_k + \xi_1) = 1] - p_k\Big| = \left|\frac{k}{2n}(p_{k-1} - p_k) + \frac{n-k}{2n}(p_{k+1} - p_k)\right| < \epsilon/2.$$

The lemma follows.■

Now it is easy to finish the proof of the theorem. It follows by Lemma 3.1 that η_k is computationally ϵ-close to $\eta_k + \xi_1$. Since variables ξ_t can be sampled in polynomial time this implies that

$$\rho(\eta_k + \xi_{t-1}, \eta_k + \xi_t) < \epsilon,$$

which implies by Proposition 2.1 that

$$\rho(\eta_k, \eta_k + \xi_t) < \epsilon t.$$

Finally, it is left to notice that the distribution of $\eta_k + \xi_t$ is statistically $me^{-\Omega(t/m)}$-close to the uniform. The theorem is proved.■

4. Applications

In this section we apply the general result of Theorem 3.1 to show the intractability of several optimization problems. Our results in this section are based on three different conjectures about the average hardness of MAXIMUM SATISFYING LINEAR SUBSYSTEM. We are unaware about efficient algorithms that refute any of these assumptions. Formally, all conjectures are independent and have different implications in the theory of approximation algorithms and in cryptography. Conjecture 2 may be of independent interest as it assumes a mixture of average case and worst-case hardness.

Problem 2 (Average-3LIN)

- INPUT: Parameters n, m and $\epsilon = \epsilon(n) > 0$.
- YES INSTANCE: A random pair (A_n, b_1), where A_n is a random $m \times n$ $(0-1)$–matrix in which every row contains exactly three ones and $b_1 \sim D_{\lceil \epsilon n \rceil}(A_n)$.
- NO INSTANCE: A random pair (A_n, b_2), where A_n is a random $m \times n$ matrix in which every row contains exactly three ones and $b_2 \sim D_{\lceil \epsilon n \rceil + 1}(A_n)$.

Conjecture 1 *For any $m = O(n)$, for any fixed $\epsilon_0 > 0$ and for any $\epsilon > \epsilon_0$ no polynomial algorithm can solve Average-3LIN with success probability greater than $1/(n \ln^2 n)$.*

Remark 1 It can be shown that if the matrix A_n happens to be "degenerate" (for example, contains two equal rows, which occurs with probability $1/n$) then one can distinguish vectors b_1 and b_2 with probability roughly $1/n$. This gives an algorithm that distinguishes (A, b_1) and (A, b_2) with success $1/n^2$. We believe that no algorithm can do substantially better than this bound, and if A_n is an expander (which occurs with probability $1 - 1/n$) then the distributions of b_1 and b_2 are indistinguishable. Thus, we could specify in Problem 2 that A is chosen uniformly from the set of good expanders and assume its $1/n^{\Omega(1)}$-intractability (but this would sacrifice the property of being samplable).

Problem 3 (Average LIN-SAT with the worst case choice of the matrix)

- INPUT: Parameters n, m and $\epsilon = \epsilon(n) > 0$.
- YES INSTANCE: Any $m \times n$ matrix A_n and a random vector $b_1 \sim D_{\lceil n^\epsilon \rceil}(A_n)$.
- NO INSTANCE: Any $m \times n$ matrix A_n and a random vector $b_2 \sim D_{\lceil n^\epsilon \rceil + 1}(A_n)$.

This problem resembles the question how to decode a linear error correcting code from n^ϵ errors. However, the matrix A_n is *not necessarily* a generator of a good code. Thus, a priori Problem 3 may be more difficult than the unique decoding of the random codeword with n^ϵ errors.

Conjecture 2 *For any $m = O(n)$, any fixed ϵ_0 and any $\epsilon > \epsilon_0$ no polynomial algorithm solves Problem 3 with success better than $1/n^{\Omega(1)}$.*

Note that this conjecture combines the worst case assumption (in the choice of A_n) with the average case assumption in the random choice of n^ϵ unsatisfied equations. Finally for our cryptographic applications in Section 4.4 we need the following conjecture.

Problem 4 (Average-NEAREST-CODEWORD)

- INPUT: Parameters n, m and $\delta = \delta(n) > 0$.
- YES INSTANCE: A random pair (A_n, b_1), where A_n is a random $m \times n$ matrix and $b_1 \sim D_{\lceil n^\delta \rceil}(A_n)$.
- NO INSTANCE: A random pair (A_n, b_2), where A_n is a random $m \times n$ matrix and $b_2 \sim D_{\lceil n^\delta \rceil + 1}(A_n)$.

Conjecture 3 *For any $m = O(n)$ there exist $\delta_1 < \delta_2 < 1/2$, s.t. for any $\delta_1 < \delta < \delta_2$ no polynomial time algorithm solves Problem 4 with success better than $1/n^{\Omega(1)}$.*

4.1. Average MAX-3LIN and Feige's Refute-3SAT hypothesis

In his paper on average case hardness versus hardness of approximation [Fei02], Feige assumes that the following problem (that we call Refute-3SAT) is hard on average and infers the hardness of approximation for several interesting problems, for which it is not known whether the approximation is NP-hard.

Problem 5 (Refute-3SAT)

- INPUT: Parameters n, m and $\epsilon > 0$.
- YES INSTANCE: 3CNF φ with n variables and m clauses, for which at least $(1-\epsilon)m$ clauses are satisfiable.
- NO INSTANCE: A random 3CNF φ with n variables and m clauses.

Note, that since nothing is said about the distribution of *yes* instances it is assumed to be chosen in the worst case. Thus in order to solve this probabilistic promise problem the algorithm should always output "yes" on every *yes* instance and w.h.p. say "no" on every *no* instance. Stated in our terms, Hypothesis 2 in [Fei02] assumes that no polynomial algorithm can solve Refute-3SAT with success $1 - o(1)$. One can similarly define Refute-3LIN problem:

Problem 6 (Refute-3LIN)

- INPUT: Parameters n, m and $\epsilon > 0$.
- YES INSTANCE: A set of m linear constraints on three variables, such that at least $(1-\epsilon)m$ constraints are satisfiable.
- NO INSTANCE: A set of m linear constraints on three variables, each of which is chosen at random from the set of all constraints over n variables.

It was shown in [Fei02] that Refute-3SAT and Refute-3LIN are essentially equivalent, and if they are hard then the following problems cannot be approximated within some constant: Min Bisection, Dense k-subgraph, Max Bipartite Clique, 2-Catalog Segmentation. Below we show that if Average-3LIN (with success probability $1/(n \ln^2 n)$) is hard for polynomial algorithms then Refute-3LIN is hard (and hence the other hardness results hold too, see Figure 1). This result may be interesting as

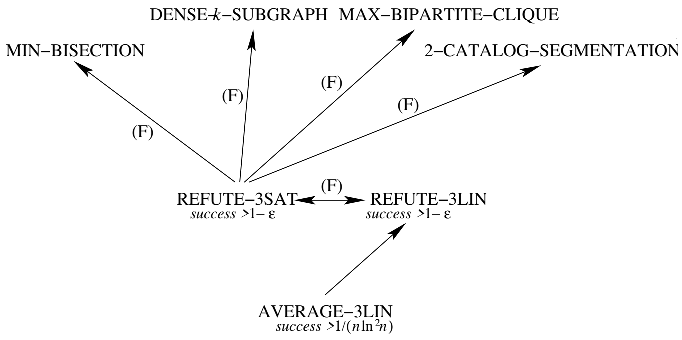

Figure 1: The graph of average-case reductions

- it gives more evidence that Hypothesis 2 in [Fei02] is true.
- Average-3LIN is a samplable problem, thus more natural from the cryptographic point of view.

Theorem 4.1 *Conjecture 1 implies that no polynomial algorithm can solve Refute-3LIN with success* $1 - o(1)$.

Proof. Assume for the sake of contradiction that there exists a polynomial algorithm $\mathcal{A}$, which always outputs "yes" on any system of m linear constraints that contains a satisfiable subsystem of size $(1-\epsilon)m$ and w.h.p. outputs "no" on the completely random system. Let $k = \epsilon m$. Then w.h.p. (according to the choice of A_n) this algorithm distinguishes between the distribution $D_k(A_n)$ and the uniform one. By Theorem 3.1 this implies that there exists an algorithm that distinguishes $D_k(A_n)$ and either $D_{k-1}(A_n)$ or $D_{k+1}(A_n)$ with success greater than $1/(n \ln^2 n)$ (to see this choose in the statement of the theorem $t = n \ln^2 n/10$). By Proposition 2.2 there exists an algorithm that solves Average-3LIN promise problem with success $1/(n \ln^2 n)$. This however contradicts to Conjecture 1.■

4.2. Nearest Codeword

In this section we study the limitations on approximability of the following problem.

Problem 7 (Nearest Codeword)

- INSTANCE: A linear binary code given by its generator $m \times n$ matrix A and a vector b.
- SOLUTION: A vector $x \in \{0,1\}^n$ that specifies a codeword Ax.
- OBJECTIVE FUNCTION: The hamming distance $d(Ax, b)$.

The best known NP-hard lower bound on the factor of approximation for the Nearest Codeword is $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$ due to [ABSS97].

Theorem 4.2 *Conjecture 2 implies that the Nearest Codeword is hard to approximate within* $n^{1-\epsilon}$.

Proof.

Lemma 4.1 *For any $m \times n$ matrix A s.t. $m > 2n$ the hamming distance of the uniformly distributed vector $y \in \{0,1\}^m$ and $\mathrm{Im}(A)$ is greater than $m/10$ w.h.p.*

Proof. By a simple counting argument. The number of different vectors in $\mathrm{Im}(A)$ is at most 2^n, the ball of radius $m/10$ contains $\binom{m}{m/10}$ points, the union of these balls covers at most

$$2^n 2^{\mathrm{H}(1/10)m} < 2^{0.97m}$$

points, which consists a negligible part of the whole space.■

The rest of the proof is similar to that of Theorem 4.1. Let $k = n^\epsilon$. By Lemma 4.1 any algorithm that approximates the Nearest Codeword can be used to distinguish between distributions $D_k(A)$ and U_m for any matrix A with success $1 - o(1)$. By Theorem 3.1 this in turns implies that $D_k(A)$ is computationally distinguishable from either $D_{k+1}(A)$ or $D_{k-1}(A)$, which contradicts to Conjecture 2, because Proposition 2.2 also holds for Average LIN-SAT problem.■

4.3. Matrix Rigidity

Recall that for $(0-1)$–matrix M its rigidity $\mathcal{R}_M(r)$ is the minimal number of entries of M that have to be changed to reduce its rank below r. Consider the following approximation problem.

Problem 8 (Approximating Matrix Rigidity)

- INPUT: Parameters m, $\epsilon, \delta > 0$.
- YES INSTANCE: Any $m \times m$ matrix M for which

$$\mathcal{R}_M(\epsilon m) < m^{1+\delta}$$

- NO INSTANCE: A random $m \times m$ matrix M.

Theorem 4.3 *Conjecture 2 implies the hardness of Problem 8 with success $1/n^{\Omega(1)}$.*

Proof. Choose $n = \epsilon m$ and denote $k = m^{1+\delta}$. We use the result of Theorem 3.1, which implies that in the assumption of Conjecture 2 it is computationally hard to distinguish distributions U_m and $D_k(A)$, for at least one $m \times n$ matrix A for every n. For $t = 0..m$ define a random matrix $M^t = (M_1|M_2|...|M_m)$ whose first t columns M_i are chosen independently with distribution $D_k(A)$ and last $m - t$ columns are chosen uniformly at random. As a corollary of Theorem 3.1 we get

Proposition 4.2 *Conjecture 2 implies that for any $t = 1..(m-1)$ it is computationally hard to distinguish M_t and M_{t+1}.*

Indeed, since the distribution $D_k(A)$ as well as U_m is samplable if an algorithm can distinguish M_t and M_{t+1} then it can also distinguish $D_k(A)$ and U_m.

Lemma 4.3 $\mathcal{R}_{M_m}(\epsilon m) < m^{1+\delta}$ *with probability one.*

Proof. Every column in M_m can be represented as a sum of a vector in $\mathrm{Im}(A)$ and a vector of weight k. Thus we can flip km entries in M_m to decrease its rank to $\dim(\mathrm{Im}(A)) = \epsilon m$.■

By Proposition 2.1 it is hard to distinguish M_0 and M_m. The former is the completely random matrix, the latter by Lemma 4.3 has low rigidity. Theorem 4.3 is proved.■

4.4. Two public-key criptosystems

We have seen in Section 3 that any of our conjectures implies the existence of a simple pseudorandom generator. Below we give two constructions of another cryptographic primitive: public key criptosystem. Very informally, this primitive is a function $f_s(x)$ which is hard to invert on average without knowledge of the "secret" s, but easy to invert given s. The first cryptosystem is easier to analyze, however it encodes only a single bit. The second system can encode up to $\Omega(n)$ bits. Both systems are secure unless Conjecture 3 is false.

Cryptosystem 1. Let $k = n^{1/2-\epsilon}$, $m = 2n$.

Generation of public/private keys.
Generate a random $m \times n$ $(0-1)$-matrix A. Generate a random vector $b \in \{0,1\}^m$ within distance k from the image of A: $b = Ax + e$, where $x \sim U_n$ and $e \in \binom{m}{k}$ is a random vector of weight k. Let $A_1 = (b|A)$ be $m \times (n+1)$ matrix that results from A by adding the column b.
The *public key* is the matrix A_1. The *private key* is the pair (A_1, e).

Encryption.
The encryption of *one* is a uniform random vector $\xi^1 \sim U_m$.
The encryption of *zero* is a random vector ξ^0 generated as

$$\xi^0 = y + e',$$

where y is a random element of the dual code with parity check matrix A_1^T (i.e. $y \in_U \mathrm{Ker}(A_1^T)$) and $e' \in \binom{m}{k}$ is a random vector of weight k.

Decryption.
For a vector $\xi \in \{0,1\}^m$ let $\delta = e^T \xi$. If δ is 0 then output *zero*. Otherwise output *one*.

The correctness and security of this cryptosystem are provided by the following theorem.

Theorem 4.4
Part 1. *The decryption algorithm returns* zero *on the encrypted zero-message with probability* $1 - o(1)$. *It returns* one *on the encrypted one-message with probability* $1/2$.

Part 2. *Conjecture 3 implies that the distributions of* (ξ^1, A_1) *and* (ξ^0, A_1) *are computationally indistinguishable.*

Before we give the proof of this theorem we note that one can stretch out the length of messages arbitrarily as well as to decrease the error in the decryption using the standard Shannon theory of error correction over the noisy channel.

Proof.
Part 1. Obviously, the probability that $e^T\xi^1 = 0$ is exactly 1/2. To estimate the probability that $e^T\xi^0 = 0$ notice that any vector y in $\mathrm{Ker}(A_1^T)$ satisfies $e^T y = 0$ since e^T belongs to the span of the rows of A_1^T. Thus for $\xi^0 = y + e'$ holds

$$e^T\xi^0 = e^T(y + e') = e^T e'.$$

Both vectors e, e' have weight k. The probability that they have a common one is less than $(1 - k/n)^k = o(1)$.

Part 2. Let us introduce intermediate random variables $\hat{A}_1 \in_U \{0,1\}^{m\times(n+1)}$ and $\hat{\xi^0} = \hat{y} + \hat{e}'$, where $\hat{y} \in_U \mathrm{Ker}(\hat{A}_1)$ and $\hat{e}' \in_U \binom{m}{k}$.

Theorem 3.1 and Conjecture 3 imply that distributions of A_1 and $\hat{A}_1$ are computationally indistinguishable, hence the distributions of (ξ^0, A_1) and $(\hat{\xi^0}, \hat{A}_1)$ are indistinguishable too. On the other hand, the kernel of $\hat{A}_1$ as any linear space can be specified as an image of linear operator B so $\hat{\xi^0}$ is chosen as

$$\hat{\xi^0} = B \cdot z + \hat{e}',$$

where B is a random $m \times (m - n - 1)$ matrix, $z \in_U \{0,1\}^{m-n-1}$ and $\hat{e}' \in \binom{m}{k}$ is a random vector of weight k. Once again we apply Theorem 3.1 and Conjecture 3 to conclude that it is computationally hard to distinguish $(\hat{\xi^0}, \hat{A}_1)$ and $(\xi^1, \hat{A}_1)$. Finally, it is hard to distinguish $(\xi^1, \hat{A}_1)$ and (ξ^1, A_1) because A_1 and $\hat{A}_1$ are computationally indistinguishable. The theorem now follows by Proposition 2.1.■

Cryptosystem 2.
Let $m = 2n$, $k = n^{1/2-\epsilon}$ and $H \in \{0,1\}^{m/10\times m}$ be a parity check matrix of any asymptotically good error correcting code for which there exists an efficient decoding algorithm.

Generation of public/private keys.
Choose a random $m \times n$ $(0-1)$–matrix A. Choose a random $n \times m$ $(0-1)$–matrix X and a random $m \times m$ matrix E, in which every row contains exactly k ones. Let $M = AX + E$. Repeat this procedure until M is not degenerate. Denote by $V = \mathrm{Ker}(A^T M^{T^{-1}}) \cap \mathrm{Ker}(H)$, let $r = \dim(V)$, assume w.l.o.g. that r is even. Choose a random partition

$$V = V_1 \oplus V_2,$$

s.t. $\dim(V_1) = \dim(V_2) = r/2$.
The *public key* is (M, A, V_1, V_2). The *private key* is (M, A, V_1, V_2, E).

Encryption.
We identify the messages of length $r/2$ (clearly r is linear in n) with vectors in V_0. To encode a message choose the corresponding $v_0 \in V_0$ and compute

$$\xi(v_0) = M^{T^{-1}}(v_0 + v_1) + e',$$

where $e' \in \binom{m}{k}$ is a random vector of weight k and $v_1 \in_U V_1$.

Decryption
To decrypt the vector ξ, compute

$$\hat{v} = E^T\xi.$$

Apply the efficient decoding algorithm for H to find the nearest codeword v to $\hat{v}$. Output the projection of v onto V_0 as the decrypted message.

Theorem 4.5
Part 1. *The decryption of the second cryptosystem is correct w.h.p.*
Part 2. *Conjecture 3 implies that the second cryptosystem is secure against the passive attack in the sense that for any two distinct messages* $v_0^1 \neq v_0^2$ *the distributions of*

$$(\xi(v_0^1), M, A, V_0, V_1) \text{ and } (\xi(v_0^2), M, A, V_0, V_1)$$

are computationally indistinguishable.

Proof.
Part 1. We write

$$E^T\xi(v_0) - (v_0 + v_1) = E^T({M^T}^{-1}(v_0 + v_1) + e') - M^T {M^T}^{-1}(v_0 + v_1) =$$

$$= (E^T - M^T){M^T}^{-1}(v_0 + v_1) + E^T e' = X^T A^T {M^T}^{-1}(v_0 + v_1) + E^T e' = E^T e'.$$

Since every row of E^T contains exactly k ones, e' contains exactly k ones and they are independent with high probability the hamming weight of $E^T e'$ is $o(n)$. Thus, w.h.p. the efficient decoding algorithm for H correctly recovers the codeword $v_0 + v_1$ and the decryption procedure succeeds.

Part 2. Let $\xi^1 \in_U \{0,1\}^m, \xi^0 = {M^T}^{-1} v_1 + e'$. Clearly it is sufficient to show that

$$(\xi(v_0^1), M, A, V_0, V_1) \text{ and } (\xi(v_0^2), M, A, V_0, V_1)$$

are both computationally indistinguishable from (ξ^1, M, A, V_0, V_1). To show this, it is sufficient in turn to prove that (ξ^0, M, A, V_0, V_1) is computationally indistinguishable from (ξ^1, M, A, V_0, V_1).

Let $\hat{A}$ be a random $m \times n$ matrix. Let $\hat{M}^T \in_U GL(\mathrm{GF}_2)$ be a random nondegenerate matrix, and similarly to the construction of public/private keys define

$$\hat{V} = \mathrm{Ker}(\hat{A}^T {\hat{M}^T}^{-1}) \cap \mathrm{Ker}(H),$$

a random partition $\hat{V} = \hat{V}_0 \oplus \hat{V}_1$, $\hat{v}_1 \in_U \hat{V}_1$ and finally

$$\hat{\xi}^0 = {\hat{M}^T}^{-1} \hat{v}_1 + e'.$$

Lemma 4.4 *Conjecture 3 implies that* (ξ_0, M, A, V_0, V_1) *and* $(\hat{\xi}_0, \hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$ *are computationally indistinguishable.*

Proof. In the same way as in the proof of Theorem 4.3 it can be shown that in the assumption of Conjecture 3 the distribution of the matrix $M = AX + E$ (where $X \in_U \{0,1\}^{n \times m}$ and E is a random $m \times m$ matrix in which every column has k elements) is computationally indistinguishable from the uniform. If we plug in the randomly chosen non-degenerate $\hat{M}$ instead of M in the construction of ξ_0 then we get exactly the variable $\hat{\xi}_0$. Thus (ξ_0, A, V_0, V_1) and $(\hat{\xi}_0, \hat{A}, \hat{V}_0, \hat{V}_1)$ are also indistinguishable. The lemma follows.■

Our next goal is to show that $(\hat{\xi}^0, \hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$ is computationally close to $(\xi^1, \hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$. For this we need to describe the distribution of ${\hat{M}^T}^{-1} \hat{v}_1$. Recall that $\hat{V}$ is the intersection of linear spaces $\mathrm{Ker}(\hat{A}^T {\hat{M}^T}^{-1})$ and $\mathrm{Ker}(H)$. The image of this space under ${\hat{M}^T}^{-1}$ is

$${\hat{M}^T}^{-1}(\hat{V}) = \mathrm{Ker}(\hat{A}^T) \cap \mathrm{Ker}(H\hat{M}^T).$$

Lemma 4.5 *1). Assume that* $H_1 \in \{0,1\}^{n_1 \times m}$ *and* $H_2 \in \{0,1\}^{n_2 \times m}$ *are independent random matrices and* $L = \mathrm{Ker}(H_1) \cap \mathrm{Ker}(H_2)$. *Let* L' *be a random linear space of dimension* $m - n_1 - n_2$ *and* $H(L', n, m)$ *be a random matrix uniformly chosen from all* $n \times m$ *matrices* H' *for which* $L' \subseteq \mathrm{Ker}(H')$. *Then the statistical distance*

$$\rho\Big((H_1, H_2, L), (H\ (L', n_1, m), H(L', n_2, m), L')\Big) < 2^{-\Omega(m-n_1-n_2)}.$$

2). Assume that $H \in n \times m$ *is a fixed given matrix of rank* n, M *is a random nondegenerate* $m \times m$ *matrix and* $H_1 = HM$. *Let* H_1' *be a random* $n \times m$ *matrix and* $M(H_1')$ *be a random nondegenerate matrix* M' *s.t.* $H_1' = HM'$. *Then*

$$\rho((M, H_1), (M(H_1'), H_1')) < 2^{-\Omega(m-n)}.$$

Proof.
1). L is a kernel of the matrix $\binom{H_1}{H_2}$, thus w.h.p. it is a random linear space of dimension $m - n_1 - n_2$. For any fixed L the conditional distribution of $H_1|(L \subseteq \mathrm{Ker}(H_1))$ and $H_2|(L \subseteq \mathrm{Ker}(H_2))$ coincides with $H(L, n_1, m)$ and $H(L, n_2, m)$ resp.
2). It is clear that H_1 is the random $n \times m$ matrix of the maximal rank. W.h.p. $\mathrm{rk}(H') = n$ and the conditional distribution of $M'|(MH = H')$ coincides with that of $M(H')$.∎

By Lemma 4.5 the image

$$L = \hat{M^T}^{-1}(\hat{V}) = \mathrm{Ker}(\hat{A}^T) \cap \mathrm{Ker}(H\hat{M}^T)$$

is a random linear space of dimension r, moreover given a random L' one can define M' and A' so that the distributions of (L, M, A) and (L', M', A') are statistically close. Since V_1 is a random $r/2$-dimensional subspace of V its image under $\hat{M^T}^{-1}$ is the random subspace $L_1 \subset L$, $\dim(L_1) = r/2$. This subspace can be represented as $L_1 = \mathrm{Im}(B)$, thus $\hat{\xi}^0 = Bz + e'$, where B is a random $r/2 \times m$ matrix, $z \in_U \{0,1\}^{r/2}$ and e' is a random vector of weight k.

We apply the result of Theorem 3.1 and Conjecture 3 once again to show that $(\hat{\xi}^0, \hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$ and $(\xi^1, \hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$ are computationally indistinguishable. Finally, $(\xi_1, \hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$ and (ξ_1, M, A, V_0, V_1) are computationally indistinguishable because so are (M, A, V_0, V_1) and $(\hat{M}, \hat{A}, \hat{V}_0, \hat{V}_1)$. The theorem follows by Proposition 2.1.∎

5. Conclusion and open problems

We have shown several relations between the average case and the worst case complexity. Our results are based on unproved hardness assumptions that are much stronger than P≠NP. However since our final goal is not to design a cryptosystem secure on practice but to better understand the complexity of several important problems, we believe that it does have sense to study such reductions. In particular it would be very interesting to see any positive algorithmic results on Problems 2–4 as well as to obtain any other reductions that would yield more information on the average complexity of these problems.

Our planted construction in Average-3LIN problem can be used as a challenge for empirical SAT solving algorithms. In particular, we believe that any SAT heuristic that doesnot invoke gaussian elimination as a subroutine will fail on this example even if we donot add any noise and the overdetermined linear system is satisfiable. Note that in the case of randomly planted SAT assignment there are non-trivial algorithms that solve it if the density is sufficienly large ([Fla03]).

We are also interested in the following question that can be considered as a step toward explicit lower bounds: construct in polynomial time a sequence (A_n, b_n) s.t. $A_n \in \{0,1\}^{m \times n}$ is expander in which every row contains finitely many ones and b_n is a vector for which the system $A_n x = b_n$ has at most $2/3m$ satisfiable equations. In other words, construct any explicit sequence that do not belong to the image of pseudorandom generator (cf. [ABRW00]).

Finally, by the analogy with Tseitin tautologies for propositional calculus, one can define Pseudo-Tseitin tautologies, which state that a given linear system is unsatisfiable in the strong sense: there is no satisfiable subsystem that contains almost all linear equations. Formalized as in [ABRW00] Pseudo-Tseitin tautologies may be a curious candidate for proving lower bounds in propositional calculus.

6. Acknowledgments

We would like to thank Amir Shpilka for introducing to us the notion of matrix rigidity. We are also grateful to Madhu Sudan for helpful discussions.

References

[ABRW00] M. Alekhnovich, E. Ben-Sasson, A. Razborov, and A. Wigderson. Pseudorandom generators in propositional complexity. In *Proceedings of the 41st IEEE FOCS*, 2000. Journal version to appear in *SIAM Journal on Computing.*

[ABSS97] S. Arora, L. Babai, J. Stern and Z. Sweedy. Hardness of Approximate Optima in Lattices, Codes, and Linear Systems. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[AD97] M. Ajtai and C. Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In *Proc. on 29th Annual ACM Symposium on Theory of Computing*, 284–293, 1997.

[ALMSS98] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof Verification and Hardness of Approximation Problems. *Journal of ACM, 45(3):501-555*, 1998.

[AS92] S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of ACM*, 45(1):70–122, 1998.

[BHR03] E. Ben-Sasson, P. Harsha and S. Raskhodnikova. Some 3CNF Properties are Hard to Test. To appear in *Proc. on 35th Annual ACM Symposium on Theory of Computing*, 2003.

[Fei02] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. on 34th Annual ACM Symposium on Theory of Computing*, 534–543, 2002.

[Fla03] A. Flaxman. A spectral technique for random satisfiable 3CNF formulas. In *Proc. on 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[Fri93] J. Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993.

[GC89] D. Gollman and W. Chambers. Clock-controlled shift registers: a review. *IEEE Journal on Selected Areas in Communications* (4) 7: 525–533, 1989.

[GG98] O. Goldreich, S. Goldwasser. On the Limits of Non-Approximability of Lattice Problems. In *Proc. on 30th Annual ACM Symposium on the Theory of Computing*, 1–9, 1998.

[Has88] J. Håstad. Dual Vectors and Lower Bounds for the Nearest Lattice Point Problem. *Combinatorica* 8 (1):75–81, 1988.

[Has01] J. Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001.

[KR98] B. Kashin and A. Razborov. Improved lower bounds on the rigidity of Hadamard matrices. *Mathematical Notes*, 63(4):471–475, 1998.

[Lok01] S. Lokam. Spectral Methods for Matrix Rigidity with Applications to Size-Depth Trade-offs and Communication Complexity. *J. of Computer and System Sciences*, 63(3): 449-473, 2001.

[LLS90] J. Lagarias, H. Lenstra, C. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica* 10(4): 333–348, 1990.

[M78] R. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. Deep Space Network Progress Report 42-44, Jet Propulsion Lab., California Institute of Technology, 114 – 116, 1978.

[NN93] J. Naor, M. Naor. Small-Bias Probability Spaces: Efficient Constructions and Applications. *SIAM J. Computing* 22(4): 838-856, 1993.

[PV91] P. Pudlak and Z. Vavřin. Computation of rigidity of order n/r for one simple matrix. *Comm. Math. Univ. Carol.*, 32(2):213–218, 1991.

[PV01] I. Pak and V. Vu. On mixing of certain random walks, cutoff phenomenon and sharp threshold of random matroid processes. *Discrete Applied Math.* 110: 251-272, 2001.

[Raz89] A. Razborov. On rigid matrices. Manuscript (in Russian), 1989.

[RR97] A. Razborov and S. Rudich. Natural Proofs. J. of Computer and System Sciences 55(1): 24–35, 1997.

[SSS97] M. Shokrollahi, D. Spielman and V. Stemann. A Remark on Matrix Rigidity. *Information Processing Letters*, 64(6): 283–285, 1997.

[Tse68] Г. С. Цейтин. О сложности вывода в исчислении высказываний. In А. О. Слисенко, editor, *Исследования по конструктивной математике и математической логике,* II*; Записки научных семинаров ЛОМИ, т. 8*, pages 234–259. Наука, Ленинград, 1968. Engl. translation: G. C. Tseitin, On the complexity of derivations in propositional calculus, in: *Studies in mathematics and mathematical logic, Part II*, ed. A. O. Slissenko, pp. 115-125.

[Val77] L. Valiant. Graph-Theoretic Arguments in Low-Level Complexity. In *Proc. 6th Symposium on Mathematical Foundations of Computer Science*: 162–176, 1977.