6.893 Approximability of Optimization Problems November 15, 1999

Lecture 18

Lecturer: Madhu Sudan *Scribe: Adam Smith*

In this lecture, we show that approximating POLYNOMIAL CONSTRAINT SATISFIABILITY (PCS) to within a constant factor is hard.

Recall the definition of PCS:

Instance A finite field F, a degree d and a collection of constraints $C_1, \ldots, C_t$.

Each constraint is a set of k points in F^m together with a small circuit constraining the values of p at those points, i.e. $C_j = (A_j, x_1^{(j)}, \ldots, x_k^{(j)})$ where $x_i^{(j)} \in F^m$ and $A_j : (F^m)^k \longrightarrow \{0,1\}$ is described by a circuit of size at most s.

The circuits consist of two kinds of wires, one for field elements and one for boolean values. The output of a circuit must be a boolean value. A gate in this circuit can be a field operation $(+, -, *, /)$, a test $(=?0)$ or a boolean operation (AND, OR, NOT).

In most instances, the parameters $|F|, m, d, k, s$ are in $\text{polylog}(t)$, and t is in $\text{poly}(n)$, where n will be defined later.

Feasible set Polynomials $p : F^m \longrightarrow F$ of total degree d.

Objective Maximize number of satisfied constraints. C_j is satisfied by p if

$$A_j(p(x_1^{(j)}), \ldots, p(x_k^{(j)})) = \text{“true”}.$$

We will prove that deciding whether the maximum number of satisfied clauses is either t or below $t/2$ is in fact NP-hard. To do this, we will first prove hardness for a related problem, POLYNOMIAL-EVOLUTION (PE).

Remark Similar ideas to the ones in this lecture give a proof that $IP = PSPACE$.

1 Polynomial evolution

A *polynomial construction rule* R over F is a circuit which takes an oracle for a polynomial p in m variables as input, and returns a new polynomial $q = R^p$ in m' variables, defined by $q(\vec{x}) \triangleq R^p(\vec{x})$. The idea is to consider a sequence of such rules and ask if when composed they can produce the zero polynomial. The resulting problem is POLYNOMIAL EVOLUTION:

Instance Polynomial construction rules $R_0, R_1, \ldots, R_l$ over F.

Feasible set Polynomials $p_0 : F^m \longrightarrow F$ such that putting $p_i = R_i^{p_{i-1}}$ for all i gives $p_l \equiv 0$ (has value 0 on its entire domain).

As long as $|F|^{m'}$ is polynomial in the description of the instance (thus roughly in $s \cdot t$), then this is indeed an NP problem.

1.1 Hardness of PE

Here we prove that PE is in fact NP-hard (hence complete) by reduction from 3-SAT and give some useful properties of the reduction.

Let ϕ be a 3-SAT formula in n variables. Choose $H \subset F$ such that $\{0,1\} \subset H$ ([1]) and $|H|^m = n$ (of course, when n isn't a power of the correct magnitude we will have round up). We can now identify H^m with $[n] = \{1, \dots, n\}$.

Any assignment $x_1 = a_1, \dots, x_n = a_n$ can be written as a function $A : H^m \longrightarrow \{0,1\} \subset F$ (that is, put $x_i = A(i)$ for $i \in [n]$). By interpolation, we can find a polynomial $\widehat{A} : F^m \longrightarrow F$ in m variables which has degree at most $|H|$ in each variable and such that $\widehat{A}\Big|_{H^m} = A$.

Conversely, any polynomial p_0 in m variables can be interpreted as an "assignment" of field values ($\in F$) to the literals $x_1, \dots, x_n$ by assigning $x_i = p_0(i)$. To interpret this as a boolean assignment, we may assign 0 to "false", 1 to "true", and all other values to "I don't know".

Based on ϕ we will construct a sequence of polynomial contruction rules which transform a given input polynomial p_0 to the zero polynomial iff the corresponding assignment satisfies ϕ ([2]). The first rule takes an m variables polynomial (representing an assignment) and outputs a $m' = 3m+3$ variable polynomial which is 0 on $H^{m'}$ iff the assignment satisfies ϕ. The m' other rules "amplify" this new polynomial so that it is 0 on a bigger and bigger set. The final polynomial will be identically zero iff the original one was zero on $H^{m'}$. Note that in fact only the first rule actually depends on ϕ.

1.2 First rule: encoding ϕ

To construct the transformation, we encode ϕ as a function

$$\phi' : H^{m'} \longrightarrow \{0,1\}.$$

Since $m' = 3m+3$, we can think of the domain as $[n]^3 \times H^3$. A point $(i_1, i_2, i_3, b_1, b_2, b_3)$ in the domain can be thought of in turn as a 3-SAT clause, where i_1, i_2, i_3 give the indices of the literals and b_1, b_2, b_3 indicate whether or not the literals are negated (we only consider cases where $b_1, b_2, b_3 \in \{0,1\}$):

$$\phi'(i_1, i_2, i_3, b_1, b_2, b_3) = \begin{cases} 1 & \text{if } b_1, b_2, b_3 \in \{0,1\} \text{ and the clause} \\ & \quad (x_{i_1} = b_1) \vee (x_{i_2} = b_2) \vee (x_{i_3} = b_3) \text{ is in } \phi \\ 0 & \text{otherwise} \end{cases}$$

Here we think of the literal x_i (resp. $\bar{x}_i$) as the predicate $x_i = 1$ (resp. $x_i = 0$).

[1] We don't need to use $\{0,1\}$; any embedding of {"true", "false"} will do.

[2] An assignment which includes "I don't know" can still satisfy ϕ if those variables which *are* set to values are sufficent to guarantee the satisfaction of all clauses.

We then construct a polynomial $\widehat{\phi}$ in m' variables of degree at most $|H|$ in each variable such that $\widehat{\phi}\Big|_{H^{m'}} = \phi'$.

The first rule, R_1, is simple: given input p_0, put

$$\begin{aligned} p_1(x_1, \dots, x_{3m+3}) \;\triangleq\; & \widehat{\phi}(x_1, \dots, x_{3m+3}) \cdot \Big(p_0(x_1, \dots, x_m) - x_{3m+1}\Big) \cdot \\ & \Big(p_0(x_{m+1}, \dots, x_{2m}) - x_{3m+2}\Big) \cdot \Big(p_0(x_{2m+1}, \dots, x_{3m}) - x_{3m+3}\Big) \end{aligned}$$

When $p_0 = \widehat{A}$ for some assignment A and the input to p_1 is in $H^{3m} \times \{0,1\}^3$, we get

$$p_1(\underbrace{i_1, i_2, i_3, b_1, b_2, b_3}_{\substack{\text{description}\\ \text{of a clause}}}) = \phi'(i_1, i_2, i_3, b_1, b_2, b_3)\Big(A(i_1) - b_1\Big)\Big(A(i_2) - b_2\Big)\Big(A(i_3) - b_3\Big)$$

This quantity is non-zero exactly when a given clause is *both* present in ϕ *and* not satisfied by A. Hence, $p_1\Big|_{H^{m'}} \equiv 0$ if and only if the input p_0 represents a satisfying assignment for ϕ. This holds even when the assignment represented by p_0 has some variables set to "I don't know."

Notice that when the input p_o is actually $\hat{A}$ for some assignment A, p_1 has degree at most $2|H|$ in each variable (since $\widehat{\phi}$ has degree at most $|H|$ in each variable).

1.3 Amplification rules

We now construct m' other rules $R_2, \dots, R_{m'+1}$ which transform p_1. They have the property that $p_{m'+1}$ is identically zero iff p_1 is zero on $H^{m'}$.

1.3.1 Univariate case

As a motivation, consider a transformation on univariate polynomials which has a similar property. Enumerate H as $\{h_1, \dots, h_{|H|}\}$. On input p, generate q defined by

$$q(r) = \sum_{j=1}^{|H|} p(h_j)\, r^j$$

Clearly, if $p(h) = 0$ for all $h \in H$, then $q \equiv 0$. Conversely, if $p\Big|_H \not\equiv 0$, then q is some non-zero polynomial of degree at most $|H|$ and so is non-zero on at least $|F \setminus H|$ points (thus, in particular, it is not identically zero).

1.3.2 Multivariate case

Now, to transform p_1, we will apply the above transformation, once in each variable. We will obtain a sequence of polynomials

$$\begin{aligned}
&p_1(x_1, x_2, \dots, x_{m'})\\
&p_2(r_1, x_2, \dots, x_{m'})\\
&\quad\vdots\\
&p_i(r_1, \dots, r_{i-1}, x_i, \dots, x_{m'})\\
&\quad\vdots\\
&p_{m'+1}(r_1, \dots, r_{m'})
\end{aligned}$$

where each transition from an x to an r follows the scheme developed for the univariate case.

For $i = 1, \dots, m'$ put

$$p_{i+1}\Big(\underbrace{\longleftarrow \vec{r} \longrightarrow}_{i-1 \text{ variables}},\ r_i\ ,\underbrace{\longleftarrow \vec{x} \longrightarrow}_{m'-i \text{ variables}}\Big) = \sum_{j=1}^{|H|} p_i\Big(\leftarrow \vec{r} \rightarrow,\ h_j\ , \leftarrow \vec{x} \rightarrow\Big)\ r_i^j$$

By the reasoning in the previous section, we see that

$$p_{i+1}\Big|_{F^i \times H^{m'-i}} \equiv 0 \qquad \Longleftrightarrow \qquad p_i\Big|_{F^{i-1} \times H^{m'-(i-1)}} \equiv 0$$

Note also that the degree in each variable is bounded by $|H|$.

Hence $p_{m'+1} \equiv 0$ if and only if the original input p_o represents a satisfying assignment to ϕ. This completes the reduction from 3-SAT.

Note that if we restrict the inputs p_0 to have small degree, we can actually make a stronger statement:

Lemma 1 ("PE lemma") *Suppose we have an instance of* PE *constructed from a formula ϕ as above.*

1. (Completeness) *If ϕ is satisfiable then there exists a p_o of degree at most $|H|$ in each variable such that $p_{m'} \equiv 0$. Moreover, the polynomials $p_1, \dots, p_{m'}$ have degree at most $2|H|$ in each variable.*
2. (Soundness—1) *Conversely, if $\exists p_0, \dots, p_{m'+1}$ polynomials of degree at most $2|H|$ in each variable such that*

$$Pr_{\vec{x} \in F^{m'}}\left[p_i(\vec{x}) = R_i^{p_{i-1}}\right] > \frac{3m'|H|}{|F|} \quad \text{and} \quad Pr_{\vec{x} \in F^{m'}}\Big[p_{m'+1}(x) = 0\Big] > \frac{m'|H|}{|F|}$$

then ϕ is satisfiable.

3. (Soundness—2) *More generally, if* $\exists p_0, \ldots, p_{m'+1}$ *polynomials of* total degree *at most* d *such that*

$$Pr_{\vec{x}\in F^{m'}}\left[p_i(\vec{x}) = R_i^{p_{i-1}}\right] > \frac{d + m'|H|}{|F|} \quad \text{and} \quad Pr_{\vec{x}\in F^{m'}}\left[p_{m'+1}(x) = 0\right] > \frac{d}{|F|}$$

then ϕ *is satisfiable.*

To prove this theorem, we recall the Schwartz-Zippel lemma:

Lemma 2 (SZ) *Let* p, q *be distinct polynomials of total degree at most* d*. Then the fraction of points in* F^m *for which* p, q disagree *is at least* $1 - \dfrac{d}{|F|}$.

Proof of Lemma 1 (Completeness) This comes directly from the construction.

(Soundness—1) Note that the rule R_1 increases the degree in each variable by at most $|H|$, by the construction of $\widehat{\phi}$. Furthermore, each rule R_{i+1} ($i = 1, \ldots, m'$) has the effect of changing the degree of the i-th variable to (at most) $|H|$, and not increasing the degree in any other variable. This means that the polynomials $R_i^{p_{i-1}}$ have degree bounded by $3|H|$ in each variable. By lemma 2, the bound on their probability of agreement actually implies that $p_{i+1} \equiv R_i^{p_{i-1}}$ for $i = 0, \ldots, m'$. Similarly, we get that $p_{m'+1} \equiv 0$. Consequently, p_0 represents a satisfying assignment to ϕ.

(Soundness—2) Same as above. If p_{i-1} has total degree d, then $R_i^{p_{i-1}}$ has degree at most $d + m'|H|$ (the first rule may actally increase the degree by $m'|H|$). As before, the probability bounds, along with lemma 2, force $p_{i+1} \equiv R_i^{p_{i-1}}$ for $i = 0, \ldots, m'$, and $p_{m'+1} \equiv 0$. ■

2 Hardness of PCS

To reduce 3-SAT to PCS, take a formula ϕ of length n and first convert it to an instance of PE as in the previous section. Choose $|H| \approx \log n$ and $m \approx \frac{\log n}{\log\log n}$ so that $|H|^m \approx n$. Furthermore, choose $|F| \approx \log^3 n$ so that it is both big enough for all the constructions to work but small enough so that $|F|^{m'} \in O(n^3)$ remains polynomial.

We will bundle together the polynomials $p_0, \ldots, p_{m'+1}$ of the previous lemma into single polynomial p in m' variables given by

$$p(v, \vec{x}) = \sum_{i=0}^{m'} \delta_i(v) p_i(\vec{x}).$$

where δ_i ($i = 0, \ldots, m'$) is the unique polynomial of degree m' such that

$$\delta_i(v) = \begin{cases} 1 & \text{if } v = i \\ 0 & \text{if } 0 \le v \le m' \text{ but } v \ne i \end{cases}$$

For each $\vec{x} \in F^{m'}$ we can make constraint on p given by

$$C_x = \bigwedge_{i=0}^{m'} \Big(p_{i+1}(x) = R_{i+1}^{p_i}(x)\Big) \wedge \Big(p_{m'+1}(x) = 0\Big)$$

(This constraint can be thought of as a constraint on the single polynomial p.)

We now get an instance of PCS with $t = |F|^{m'} \in O(n^3)$ constraints. Each of these circuits checks p at $k \approx (m'+1)(|H|+1) \in \text{polylog}(n)$ points and has size s of the same order as k. If we fix the degree bound to be $d = m'(2|H|+1)$, we also have the following result, which follows directly from lemma 1 (the PE lemma):

Lemma 3 *1.* (Completeness) *If ϕ is satisfiable, then $\exists p$ of degree at most d such that all constraints are satisfied.*

2. (Soundness) *Conversely, if $\exists p$ of degree d which satisfies $\frac{d+m'|H|}{|F|}\ t < \frac{2d}{|F|}$ constraints, then in fact ϕ is satisfiable.*

This shows that PCS is NP-hard to approximate to within a $\frac{2d}{|F|}$ factor. By tweaking the parameters, we can get the result to within any desired constant, in particular 1/2. Based on the PCP for PCS ([3]) that was seen in the last lecture, we get:

$$NP \subseteq PCP_{1,\frac{1}{2}}(O(\log n)\ ,\ \text{polylog}(n))$$

[3] TLA's appear everywhere in computer science