PCPs and Inapproximability
The PCP Theorem and GAP-MAX-E3SAT

My T. Thai

1 Recap

1.1 NP-Completeness

Language L is in NP iff there is a polynomial time deterministic verifier V (say a Turning machine) and an arbitrarily powerful prover P, which the following properties:

- Completeness: For every $x \in L$, P can write a proof π of length poly($|x|$) that V accepts
- Soundness: For every $x \notin L$, no matter what π proof P writes, V rejects.

For example, consider the Vertex Cover (VC) problem. Input $x = (G, k)$ where G is a graph and k is a positive integer number. (Note that here is the decision problem of VC). The question here is that if there exists a VC of G with at most k vertices. The prover P will write a proof π which is a subset S of vertices. The verifier C can verify if π is a VC or not. Thus VC is in NP.

1.2 PCP

The first question arising from the above definition is: What if we change the requirement of verifier V? What if V produces one-sided-error? That is, if x is a NO instance, V still output YES. What if V can read a certain amount of π, not the whole π as above?

- **Randomness complexity**: The total bits of randomness (usually is $O(\log n)$) that V can uses, called r. In the case of NP as above, the randomness complexity is 0.
- **Query complexity**: The total bits of π that V can read, called q. that is, V uses r bits of randomness to choose q random locations in π and reads the bits in these q randomly chosen locations.

Such a verifier V is called (r, q)-restricted verifier. More formally:

Definition 1 *V is said an (r, q)-restricted verifier V if V is a randomized poly-time algorithm with randomness complexity r and query complexity q.*

(Note that r and q can be a function of the input size, not necessary a constant.)

Given $0 \leq s \leq c \leq 1$, $\Pi \in PCP_{c,s}[r,q]$ if there exists an (r,q)-restricted verifier satisfying the following conditions:

- **Completeness:** If x is a YES-instance, then there exists a proof π such that $\text{Prob}[V(x,\pi) = YES] \geq c$
- **Soundness:** If x is a NO-instance, then for any π, $\text{Prob}[V(x,\pi) = YES] \leq s$

When $c = 1$, and $s = 1/2$, for simplicity, we write $\text{PCP}[r,q]$ instead of $\text{PCP}_{1,1/2}[r,q]$

Now, can you show that $\text{NP} = \text{PCP}_{1,s}[0, poly(n)]$ for all $s < 1$?

Can you show that $PCP[O(\log n), O(\log n)] \subseteq NP$? Note that when $r = O(\log n)$, the the proof-checking can be derandomized, that is, V can be simulated by a polynomial time deterministic verifier that simulates the computation of V on each of the $2^r = n^{O(1)}$ possible random inputs and then computes the probability that $V(x,\pi)$ accepts, then accepts if and only if this probability is one.

2 The PCP Theorem

Theorem 1 *The PCP Theorem: There exists two constant numbers c_1 and c_2 such that $NP = PCP[c_1 \log n, c_2]$. For simplicity, $NP = PCP[O(\log n), O(1)]$*

Theorem 2 *Hastad (1997) Theorem [2] For every constant $\epsilon, \delta > 0$, $NP = PCP_{1-\epsilon,1/2+\delta}[O(\log n), 3]$*

That is, for every constant $\epsilon, \delta > 0$, there is a poly-size PCP for NP that reads just three random bits and test their XOR. Its completeness is $1-\epsilon$ and its soundness is $1/2+\epsilon$. This result has imperfect completeness. However, if one is willing to allow an adaptive three-bit-querying verifier, i.e. the verifier V does not have to pick the three bits in advance but can base what bit it reads next on what it's seen so far, then one can get completeness 1. That is, $\text{NP} = \text{PCP}_{1,1/2+\delta}[O(\log n), O(1)]$. This result is due to Guruswami, Lewin, Sudan, and Trevisan [1].

3 Hardness of Approximation

Recall definition of Approximation Algorithm and Approximation Ratio.

Consider the MAX-E3SAT problem, defined as follows:

Definition 2 *(***MAX-E3SAT***) Given an E3CNF formula, i.e., a conjunction of clauses over boolean variables $x_1, x_2, \ldots, x_n$, where a clause is an OR of exactly 3 literals, x_i or $\bar{x}_i$, find an assignment (of TURE/FALSE) to the variables satisfying as many clauses as possible.*

If there is no constraint on the number of literals in each clause, we have a MAX-SAT problem, and if each clause has at most 3 literals, we have MAX-3SAT.

Let us consider the MAX-3ESAT, if we set each variable to true with probability 1/2 independently, we can obtain a 7/8-randomized approximation algorithm for MAX-E3SAT. This can be de-randomized using a well-known method, called Conditional Expectations, thus leading the a deterministic approximation algorithm. Now, the question here is that: Does there exist an $(7/8 + \epsilon)$-approximation algorithm for MAX-E3SAT where $\epsilon > 0$. This type of question is called hardness of approximation.

To answer this question, we need to consider the *gap-version* of the optimization problem Π. Now, the gap-version of MAX-E3SAT is defined as follows:

Definition 3 *GAP-MAX-E3SAT$_{c,s}$ ($0 < s \leq c \leq 1$): Given an E3SAT formula φ on m clauses and let OPT denote the maximum number of clauses in φ that are satisfied by a truth assignment.*

- *output YES if $OPT \geq cm$;*
- *output NO if $OPT < sm$*

Note that GAP-MAX-E3SAT being NP-hard means that there is a deterministic polynomial time reduction, R, from your favorite NP-complete language, for example, 3-COLOR, to E3SAT, such that

- Completeness: Given $G \in$ 3-COLOR, $R(G)$ gives an E3SAT formula with $OPT \geq cm$

- Soundness: Given $G \notin$ 3-COLOR, $R(G)$ gives an E3SAT formula with $OPT < sm$.

That is, if G is a Yes instance, R will produce a Yes Instance for GAP-MAX-E3SAT. If G is a NO instance, R will produce a NO instance for GAP-MAX-E3SAT.

(3-COLOR: Given a graph G, color the vertices with 3 colors such that no edge is monochromatic if such a coloring exists.)

Remark. GAP-MAX-E3SAT$_{c,s}$ being NP-hard implies that there is no polynomial time (s/c)-factor approximation algorithm for MAX-E3SAT unless P = NP. We can easily prove this as follows:

Proof. Assume we have such an algorithm, called $\mathcal{A}$ which has an approximation ratio s/c for MAX-E3SAT, we then show how to solve 3-COLOR in polynomial time. To do this, given a graph G, apply the reduction R reducing it to E3SAT and then run the algorithm $\mathcal{A}$. If $G \in$ 3-COLOR, then this will produce an assignment that satisfies at least $(s/c)cm = sm$ clauses in $R(G)$. If not, the algorithm will be unable to produce an assignment that satisfies as many as sm clauses of $R(G)$. Thus we can distinguish the two cases and get a polynomial time algorithm for 3-COLOR. □

That is why we need to introduce the gap-version. To prove the hardness of approximation of some optimization problem Π, we can "simply" prove that the gap version of Π is NP-hard.

You can use a similar argument to show that such $(s/c)-$approximation algorithm for MAX-E3SAT can solve the GAP-MAX-E3SAT in polynomial time (but GAP-MAX-3ESAT is NP-hard) (Can you show it?)

(**Proof.** Suppose there exists such an algorithm $\mathcal{A}$ for MAX-E3SAT. Then consider the following algorithm $\mathcal{B}$ for GAP-MAX-E3SAT$_{c,s}$:

Algorithm $\mathcal{B}$: On input φ; Run $\mathcal{A}$ on φ and output YES if $\mathcal{A}(\varphi) \geq (s/c)cm = sm$. Output NO if $\mathcal{A}(\varphi) < (s/c)cm = sm$

Then $\mathcal{B}$ solves the GAP-MAX-E3SAT$_{c,s}$ in polynomial time. □)

Now, you are questioning: What this has anything to do with PCP?

Theorem 3 *The following two statements are equivalent*

- *The PCP theorem* $NP = PCP[O(\log n), O(1)]$
- *There exists a constant* $s < 1$ *such that GAP-MAX-E3SAT*$_{1,s}$ *is NP-hard.*

Proof. ($\Leftarrow$) We first show that the second statement implies the first. (I will write $\text{GAP}_{1,s}$ for short, instead of GAP-MAX-E3SAT$_{1,s}$). Assume that $\text{GAP}_{1,s}$ is NP-hard. We will construct a PCP system for 3-COLOR. That is, we need to construct a $(O(\log n), O(1))$-restricted verifier V for 3-COLOR. Given an instance G of 3-COLOR on n vertices, the verifier V runs the reduction from 3-COLOR to E3SAT, let φ be the constructed formula on m clauses. Note that φ has $poly(n)$ variables. V uses $\log m = O(\log n)$ random bits to choose a random clause, called ϕ, and queries a proof π on three positions (for each literal in ϕ) and check current assignment satisfies ϕ. (Note that prove P gives a proof π which is an assignment to the variables in φ.) We now show that the above PCP system has the required properties:

- Completeness: If $G \in$ 3-COLOR, then φ has $OPT = m$. In this case, prover P can write down the optimal assignment, which implies that all the clauses are satisfied, and hence V accepts with probability 1.
- Soundness: If $G \notin$ 3-COLOR, then φ has $OPT < sm$. Thus for any assignment π P provides, V picks a satisfied clause with probability less than s, that is, V accepts with probability less than s. The soundness can be brought down to 1/2 by repeating the check $O(1)$ many times independently ($\log_2(1/s)$) and output YES only iff all these times have a YES output.

Note that the total random bits is at most $\log_2(1/s) \log_2 m = O(\log |\varphi|)$ and the total query bits is at most $2 \log_2(1/s)$. Thus V is $(O(\log n), O(1)$-restricted.

(Question: To prove ($\Leftarrow$), can we construct an $(O(\log n), O(1))-$restricted verifier for $\text{GAP}_{1,s}$ instead of for 3-COLOR?)

($\Rightarrow$) Now, we prove the first statement implies the second. Assume the PCP theorem. We will give a deterministic polynomial time reduction R from 3-COLOR to $\text{GAP}_{1,s}$. From the PCP theorem, we know that there exists an (r, q)-restricted verifier V for 3-COLOR (where $r = c_1 \log n$ and $q = c_2$, c_1 and c_2 are positive constants). Consider any input G of 3-COLOR, we will use V to construct a E3SAT formula φ as an input of GAP_{1s} such that the above two conditions are satisfied. (completeness: if $G \in$ 3-COLOR, then φ has $OPT = m$. Soundness: if $G \notin$ 3-COLOR, then φ has $OPT < sm$.)

Notice that proof π has the length at most $p = poly(n)$ where $n = |G|$. We will think of the bits of the proof π as boolean variables $x_1, x_2, \ldots, x_p$ for an E3SAT formula. (That is, each variable x_i is equivalent to each bit in π and a truth assignment for these variables is a proof π.)

Given G, our R will work as follows. R will first run V. Then R enumerates all the $2^r = N$ random choices of V, each choice gives some q proof location $(x_{i1}, x_{i,2}, \ldots, x_{iq})$ such that some truth assignments on $(x_{i1}, x_{i,2}, \ldots, x_{iq})$ can make V accepts G. We can easily write a CNF ϕ on the q bits so that a truth assignment satisfies ϕ is also a truth assignment that make V accepts G. R further canonically converts $\phi(x_{i1}, x_{i,2}, \ldots, x_{iq})$ to an equivalent E3CNF formula. (In this step, R may need to add some auxiliary variables, $y_1, \ldots, y_{q'}$.) Without loss of generality, we may assume that each equivalent E3CNF has exactly K clauses where $K = q2^q$. Finally, R outputs the conjunction of all these $m = NK$ clauses, which is our φ. We now show that this reduction works. That is, it satisfies the two conditions:

(1) Completeness. If $G \in$ 3-COLOR, then we know that there exists a proof π such that $\text{Prob}[V(G, \pi) = YES] = 1$, that is, $V(G, \pi)$ always except G. Thus π is also a truth assignment satisfies all clauses in φ, thus $OPT = m$.

(2) Soundness. If $G \notin$ 3-COLOR, then for every proof π, $V(G, \pi)$ accepts G with a probability at most $1/2$. It implies that $V(G, \pi)$ can accept at most $1/2$ of N random strings. Or we can say that at least $1/2$ of N checks (random strings) must fail. Whenever a check fails, the corresponding E3CNF has at most $K - 1 = K(1 - 1/K)$ many satisfied clauses. (A CNF is not satisfied if there exists at least one clause not satisfied). Thus the total number of simultaneously satisfiable clauses is at most

$$\frac{N}{2}K(1 - \frac{1}{K}) + \frac{N}{2}K = NK(1 - \frac{1}{2K}) = m(1 - \frac{1}{2K})$$

Set $s = (1 - \frac{1}{2K})$, then $OPT < sm$ □

Question: Can you show that GAP-MAX-E3SAT$_{1,1-1/m}$ is NP-hard? Do you need the PCP theorem for this proof? Likewise, can you show that GAP-3-COLOR$_{1,1-1/m}$ is NP-hard?

From Theorem 3, proving the PCP theorem is equivalent to proving that GAP-MAX-E3SAT$_{1,s}$ is NP-hard for some constant $s < 1$. In fact, Dinur's

proof proves this version of the PCP theorem. She starts from showing that GAP3-COLOR$_{1,s}$ is NP-hard. Why the proof of PCP theorem is so difficult? It is easy to see that GAP3-COLOR$_{1,1-1/m}$ is NP-hard, however, it is not easy to amplify the soundness gap from $1/m$ to some constant $s < 1$.

References

[1] V. Guruswami, D. Lewin, M. Sudan, and L. Trevisan, "A tight characterization of NP with 3 query PCPs," FOCS 98.

[2] J. Hastad, "Some Optimal Inapproximability Resuts," J. ACM, 48(4):798–859, 2001