Probabilistically Checkable Proofs

3-SAT

- Given a Boolean formula, does there exist an assignment which satisfies it?

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4)$$

- Compute an assignment which satisfies as many clauses as possible.

- α-approximation algorithm: outputs a solution whose cost is at least α times the cost of the optimal solution.
  - How do prove cost of solution is at least α times cost of optimal solution when we don't know OPT?
  - Find suitable upper bounds OPT.

3-SAT

- OPT$\leq m$. Therefore, an algorithm producing an assignment satisfying at least αm constraints will be an α-approximation algorithm.

Algorithm: For each i, set X_i to TRUE with probability ½, and FALSE with probability ½.

- Fix any clause $(X_a \vee \neg X_b \vee X_c)$.

$$\Pr[clause\ is\ TRUE] = 1 - \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{7}{8}$$

- $E[number\ of\ clauses\ satisfied] = \frac{7}{8}m \geq \frac{7}{8}OPT$

- Therefore, this is a $7/8$-approximation algorithm.
- NP-hard to do better than this!

SAT

- Given a Boolean formula, does there exist an assignment which satisfies it.

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4)$$

- YES instance if there exists an assignment which satisfies it, else NO instance.
- If SAT instance is a YES instance, the satisfying assignment suffices to “prove” this.
- Given a polynomial sized proof, a Verifier (Turing machine) can verify in polynomial time that the instance is a YES instance.

Proof

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4)$$

- Proof is "11101", i.e., $x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ x_5 = 1.$
- Other proofs "10011",…

- A verifier (Turing Machine) can verify in polynomial time that this formula is satisfiable.

- Does the verifier need to read the whole proof, or can the verifier make a decision after reading only $O(1)$ bits of the proof?

Verifiers

- Allow verifier to be randomized. Verifier's decision should be correct with "good" probability.

- Verifier can use at most r random coins, and read q locations in the proof.
- If SAT instance is satisfiable, verifier should accept with probability at least c.
- If SAT instance is not satisfiable, verifier should accept with probability at most s.

Verifier

- Prover writes down a "proof".
- Verifier tosses r independent random coins to decide upon the q random locations $l_1, l_2, \ldots, l_q$ of the proof to query.
- Compute $g\left(X_{l_1}, \ldots, X_{l_q}\right)$ on the values at those locations.
  - $g(\cdot)$ depends on the PCP.
- Verifier accepts if $g(\cdot)$ evaluates to 1, and reject if it evaluates to 0.

PCP

- $\mathrm{PCP}_{c,s}(r,q)$ = class of languages which have a probabilistically checkable proof with these parameters.

- Want $q = O(1), r = O(\log n)$. Polynomial length proof.

- $\mathrm{PCP}_{c,s}\big(\mathrm{O}(\log n), O(1)\big) = NP$

[Arora, Safra – 92, Arora, Lund, Motwani, Sudan, Szegedy - 92],[Dinur - 04]

Hardness of Approximation

- Each value of the random coins $R \sim \{0,1\}^r$ gives q locations $l_1^{(R)}, l_2^{(R)}, \dots, l_q^{(R)}$ and a test $g\left(l_1^{(R)}, l_2^{(R)}, \dots, l_q^{(R)}\right)$. Consider the set of tests $\left\{g\left(l_1^{(R)}, l_2^{(R)}, \dots, l_q^{(R)}\right) : R\right\}$.

This can be viewed as a SAT problem:

- Variables are the entries of the proof $l_1, l_2, \dots$
- Constraints are $\left\{g\left(l_1^{(R)}, l_2^{(R)}, \dots, l_q^{(R)}\right) : R\right\}$.
- Find an assignment to the variables that satisfies as many constraints as possible.

Hardness of Approximation

- If SAT instance is satisfiable, then verifier accepts with probability at least c.
- There exists an assignment to $l_1, l_2, \ldots$ which satisfies at least c fraction of the constraints in $\left\{g\left(l_1^{(R)}, l_2^{(R)}, \ldots, l_q^{(R)}\right) : R\right\}$.

- If SAT instance is not satisfiable, then verifier accepts with probability at most s.
- Any assignment to $l_1, l_2, \ldots$ will satisfy at most s fraction of the constraints in $\left\{g\left(l_1^{(R)}, l_2^{(R)}, \ldots, l_q^{(R)}\right) : R\right\}$.

Hardness of Approximation

- Therefore, for $\left\{g\left(l_1^{(R)}, l_2^{(R)}, \dots, l_q^{(R)}\right) : R\right\}$, it is NP-hard to determine whether there is an assignment which satisfies at least c fraction of the constraints, or whether all assignments will satisfy at most s fraction of the constraints.

- Therefore, it is NP-hard to obtain any approximation algorithm with approximation factor better then s/c

Max 3-SAT

- [Hastad 01]: For every $\delta > 0$, and every $L \in NP$, there is a PCP with $q = 3$, $c \geq 1 - \delta$ and $s \leq \frac{1}{2} + \delta$. Moreover, the verifier chooses indices $(i_1, i_2, i_3) \sim [m]^3$ and $b \sim \{0,1\}$ according to some distribution and checks whether $l_{i_1} + l_{i_2} + l_{i_3} = b \ (mod\ 2)$.

- For any ϵ, obtaining $\frac{1}{2} + \epsilon$ approximation for Max-E3LIN is NP-hard.
- A random assignment gives $\frac{1}{2}$ approximation (verify).
- A reduction from Max-E3LIN to Max-3SAT shows that for any ϵ, obtaining $\frac{7}{8} + \epsilon$ approximation for Max-3SAT is NP-hard.

- Many other hardness of approximation results based on PCPs.

Unique Games

- Unique Games: Given a graph $G = (V, E)$, alphabet $[k]$, and bijections $\pi_{uv}: [k] \to [k]$ for each $\{u, v\} \in E$, compute an assignment $\sigma: V \to [k]$ that maximizes the fraction of constraints satisfied.

- If there exists an assignment which satisfies all the constraints, easy to find it.

- If there exists an assignment which satisfies at least 99% of the constraints, can we find a good assignment?

$$X_1 - X_2 = a_1 \bmod p$$
$$X_2 - X_3 = a_2 \bmod p$$
$$X_1 - X_5 = a_3 \bmod p$$
$$\vdots$$

Unique Games

$$X_1 - X_2 = a_1 \bmod p$$
$$X_2 - X_3 = a_2 \bmod p$$
$$X_1 - X_5 = a_3 \bmod p$$
$$\vdots$$

- Random assignment satisfies $1/p$ fraction of constraints.

- For each i, set X_i to be a random element in $\{0,1,\dots,p-1\}$.
- $\Pr[X_1 = X_2 + a_1 \bmod p] = \frac{1}{p}$.

- Better approximation algorithms known using semidefinite programming techniques.

Unique Games Conjecture [Khot-02]

- Conjecture: For every sufficiently small ϵ, there exists a k such that for Unique Games instances with alphabet size k, it is NP-hard to distinguish between the following two cases

1. There exists an assignment satisfying $1 - \epsilon$ fraction of constraints.
2. All assignments satisfy at most ϵ fraction of the constraints.

- Implies optimal hardness of approximation results for many problems, e.g. Max-cut, min vertex cover, CSPs, etc.

- Conjecture is still open!

Other hardness assumptions

Exponential Time Hypothesis

- [Impagliazzo, Paturi 99] $\exists \delta > 0$ such that 3SAT $\notin$ Time$\left(2^{\delta n}\right)$

- Lots of research on hardness of approximation.