Introduction

Q: Can we solve Independent Set problem optimally in polynomial time?
A: Reduce 3SAT to Independent Set
Start from 3SAT instance and construct graph G and integer k such that:

- If φ is satisfiable then G has an independent Set of size k.
- If not all independent sets in G have size at most k-1.

Introduction

Poly time exact algorithm for IS + poly time reduction

$$\Downarrow$$

Poly time exact algorithm for 3SAT + Cook's theorem

$$\Downarrow$$

Poly time exact algorithm for every problem in NP

Introduction

Unlikely to find algorithm that:

- Runs in polynomial time on all instances
- Finds optimal solutions

Approximation algorithms: Algorithms that run in polynomial time on all instances and find sub-optimal solutions.

Approximation ratio r: We say that an algorithm is r-approximate for a minimization problem if, on every input, the algorithm finds a solution whose cost is at most r times the optimum.

Introduction

Q:What is the approximability of Independent Set problem?

A: Look at the reduction from 3SAT:

- If φ has an assignment that satisfies all the clauses except c ones, then G has an Independent Set of size k-c.
- Given such an assignment, the Independent Set is easy to construct.

Q: How can we get an inapproximability result for IS?
A: We need a much stronger reduction.
Suppose we want show that no 2-approximate algorithm exists for IS problem assuming P $\neq$ NP.
Reduction with property:

- If φ satisfiable then $OPT_{IS} \geq k$
- If φ not satisfiable then $OPT_{IS} < k/2$

Given G,the 2-approximation algorithm will find a solution S with cost(S)$\geq k/2 \iff \varphi$is satisfiable

Limitations

We showed that no 2 approximation algorithm exists for IS problem assuming P $\neq$ NP.

The only inapproximability results that can be proved with such reductions are for problems that remain NP-hard even restricted to instances where the optimum is a small constant.

To prove more general inapproximability results it is necessary to first find a machine model for NP in which accepting computations would be "very far" from rejecting computations.

NP

Problem Π is in NP if there exists a poly-time verification algorithm V(or verifier) that takes two inputs: the input x of an instance of Π and some short(polynomially bounded in the length of x) proof w. If instance is:

- "Yes" instance, then there exists some short proof w that V outputs "Yes"(V accepts w).
- "No" instance, then V outputs "No" for any short proof w(V rejects all w).

PCPs

We define PCPs by considering a probabilistic modification of the definition of NP.

PCP[r(n),q(n)

Every problem Π is in PCP[r(n),q(n)] if there is an (r(n),q(n))-restricted verifier V such that if instance is:

- "Yes" instance, then there is a w such that V accepts with probability 1
- "No" instance, then for every w the probability that V accepts is at most $1/2$.

PCP

1. We say that a verifier is (r(n),q(n))-restricted if, for every input x of length n and for every w, V makes at most q(n) queries into w and uses at most r(n) random bits.
2. Every problem Π is in $PCP_{c(n),s(n)}[r(n), q(n)]$ with 0$\leq$ $s(n) < c(n)$ $\leq$1 if there is an (r(n),q(n))-restricted verifier V such that if instance is:

- "Yes" then there is a w such that V accepts with probability at least c(n).
- "No" then for every w the probability that V accepts is at most s(n).

PCPs and NP

Surprisingly,we can have a weaker,randomized concept of a verifier for any problem in NP.Instead of reading the entire proof, the verifier will only examine some number of random bits in the proof.

The verifier has very little power and yet this is enough to distinguish between "Yes" and "No" with reasonable probability!!

PCP Theorem

PCP Theorem[Arora-Lund-Motwani-Sudan-Szegedy 92]

There exists a positive constant k
such that NP $\subseteq PCP_{1,1/2}(O(\log(n)),k)$

PCP use in proving inapproximability

Idea:Given any NP-complete problem Π and a verifier V we consider all the $2^{clogn} = n^c$ possible strings that V could use.

Given one random string,in our constraint satisfaction problem we create constraint $f(x_{i_1},..x_{i_k})$, where x_i is the i bit of the proof.

By the PCP for any "Yes" instance there exists a proof such that V accepts with probability 1⇒ there is a way of setting the variables so that all the constraints are satisfiable.

Similarly, for any "No" instance,for any proof V accepts with probability$\leq 1/2 \Rightarrow$ thus for any setting of the variables at most half of the constraints can be satisfiable.

PCP use in proving inapproximability

Now suppose we have an a approximation algorithm for this maximum constraint satisfaction problem with
$a > 1/2$.
If the constraint satisfaction problem corresponds to a:

- "Yes" instance, all the constraints are satisfiable and our approximation algorithm will satisfy more than half the constraints.
- "No instance, at most half the constraints are satisfiable and our approximation algorithm will satisfy at most half the constraints.
  We can distinguish "Yes" and "No" thus P=NP.

PCP and the Approximability of MAX 3SAT

Theorem

The PCP Theorem implies that there is a an $\varepsilon>0$ such that here is no polynomial time (1-ε)-approximate algorithm for MAX-3SAT, unless P=NP.

PROOF: Let L $\in PCP[r(n), q(n)]$ be an NP-complete problem, where q is a constant and let V be the (O(log(n),q)-restricted verifier for L. Given an instance x of L we construct a 3CNF formula φ_x with m clauses such that, for some $\varepsilon>0$ to be determined,

- x$\in$ L$\Rightarrow$ φ_x is satisfiable
- x$\notin$ L$\Rightarrow$ no assignment satisfies more than (1-ε)m clauses of φ_x.

PCP and the Approximability of MAX 3SAT

For each R,V chooses q positions and accepts iff $f_R(w_{i_1}, ..w_{i_q})$=1. Simulation of possible computation of the verifier as a Boolean formula:

1. $\forall$ R add clauses that represent $f_R(x_{i_1}, ..x_{i_q})$ For a qCNF expression we need to add at most 2^q clauses.
2. Next we convert clauses of length q to clauses of length 3.
3. Overall,transformation creates φ_x with at most $q2^q$ 3CNF clauses.

Relation of optimum φ_z as an instance of MAX3SAT and the question whether z$\in$ L:

- If z$\in$ L, then there is a witness w such that V accepts for every R. Set $x_i = w_i$ and set the auxiliary variables appropriately, then the assignment satisfies all clauses and φ_z is satisfiable.
- If z$\notin$ L then consider an arbitrary assignment to the variables x_i and to the auxiliary variables,and consider the string w where w_i is set equal to x_i. The witness w makes the verifier reject for half of the R$\in \{0,1\}^{r(|z|)}$ and for each such R,one of the clauses representing f_R fails. Overall, at least a fraction $\varepsilon = 1/2q2^q$ of clauses fails.

Theorem

If there is a reduction as above for some problem L$\in$ NP,then L $\in$ PCP[O(logn),0(1)].In particular,if L is NP-complete then the PCP Theorem holds.

PROOF: We describe how to construct a verifier for L. V on input z expects w to be satisfying assignment for φ_z. V picks O$(1/\varepsilon)$ clauses of φ at random, and checks that w satisfies all of them. The number of random bits used by the verifier is O$(logm/\varepsilon)$=O$(log|z|)$. The number bits of the witness that are read by the verifier is O$(1/\varepsilon)$=O(1).

Tight inapproximability

Hastad

For every $\varepsilon > 0$, $NP = PCP_{1-\varepsilon,1/2+\varepsilon}[O(log(n)), 3]$. Furthermore the verifier behaves as follows:it uses its randomness to pick three entries i,j,k in the witness and a bit b, and accepts if and only if $w_i \oplus w_j \oplus w_k = b$.

- For every $\varepsilon > 0$ there is a reduction that given a 3CNF formula constructs a system of linear equations over GF(2) with 3 variables per equation.
- It is not possible to approximate Max E3LIN-2 within a factor better than 2 unless P=NP.

Max ELIN-2 $\leq$ Max 3SAT

Take an instance I of Max E3LIN-2 and construct an instance φ_I of Max 3SAT.

1. Transform every equation $w_i \oplus w_j \oplus w_k$=b in I into conjunction of 4 clauses.
2. φ_I is the conjuction of all these clauses.
3. Let m the number of equations in I then φ_I has 4m clauses

Max ELIN-2$\leq$ Max 3SAT

- If$\geq$m(1-ε) of the E3LIN equations could be satisfied,then $\geq$4m(1-ε) of the clauses can be satisfied using the same assignment.
- If$<$m(1/2+ε) equations are satisfied then $<$ 3.5m $+\varepsilon$m clauses satisfied.

Tight approximation for Max 3SAT

Theorem

If there is an r-approximate algorithm for Max 3SAT,where $r > 7/8$, then P=NP.

Approximation-preserving reduction

L-reduction

Given two optimization problems P and P' we say we have an L-reduction from P to P' if for some a,b$>$ 0:

1. For each instance I of P we can compute in polynomial time instance I' of P'.
2. OPT(I')$\leq$aOPT(I)
3. Given a solution of value V' to I' we can compute in polynomial time a solution of value V to I such that $|OPT(I) - V| \leq b|OPT(I') - V'|$.

Approximation- preserving reduction

Theorem

If there is an L-reduction with parameters a,b from maximization problem P to maximization problem P' and there is an r-approximation algorithm for P' then there is an (1-ab(1-r))-approximation algorithm for P.

We give an L-reduction from Max E3SAT to the maximum independent set problem.

1. Given I with m clauses we create graph with 3m nodes,one for each literal in I.
2. For any clause we add edges connecting literals in the clause
3. For any literal x_i we add an edge to $\overline{x}_i$.

The construction implies that:

- OPT(I)=OPT(I')
- $V \geq V'$

The statements above imply that we have an L-reduction with a=b=1. The following is then immediate.

Theorem

There is no r-approximation algorithm for the maximum independent set problem with $r > 7/8$, unless P=NP.

THANK YOU!!!