Approximation algorithms

Hardness of approximation (chapter 29)

NTUA, June 2009

Reductions, gaps and hardness factors

Main technical core is the PCP theorem.

Example:

We can map a Boolean formula φ to a graph *G=(V,E)* such that:

- If φ is satisfiable, *G* has a vertex cover of size $\leq \frac{2}{3}|V|$
- If φ is not satisfiable, smallest v.c. is of size $> a\frac{2}{3}|V|$

Consequence:

There is no polynomial time algorithm that achieves an approximation guarantee of *a* (unless *P=NP*).

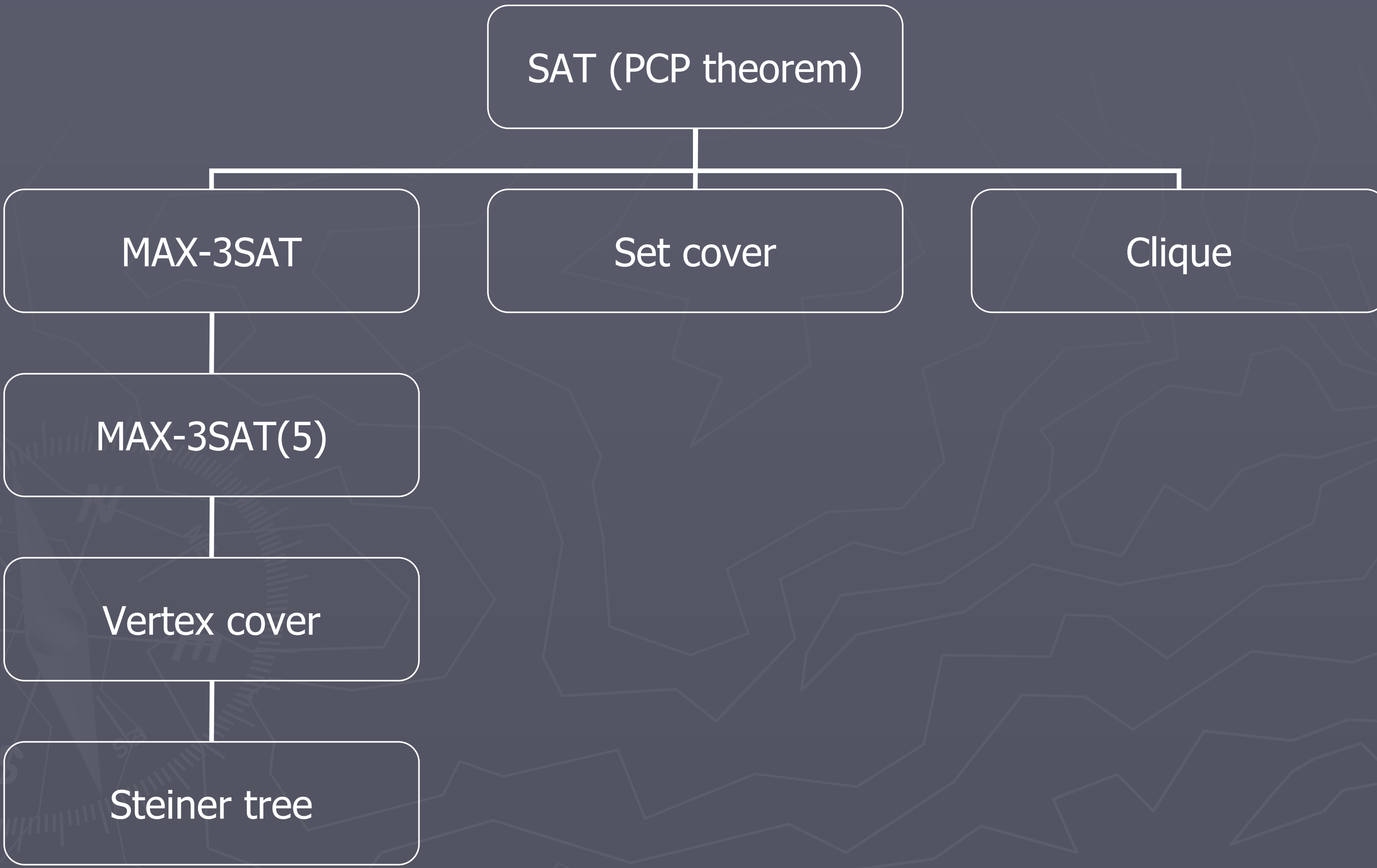
Reductions, gaps and hardness factors

Definitions

- Let *Π* be a minimization problem. A <u>gap-introducing reduction</u> comes with parameters *f* and *a*. In polynomial time maps an instance φ of SAT to an instance *x* of *Π* such that:
  - If φ is satisfiable, $OPT(x) \le f(x)$
  - If φ is not satisfiable, $OPT(x) > a(|x|)\,f(x)$

- Let Π_1 be a minimization problem and Π_2 be a maximization problem. A <u>gap-preserving reduction</u> comes with parameters f_1, *a*, f_2 and b. In polynomial time maps an instance x of Π_1 to an instance *y* of Π_2 such that:
  - $OPT(x) \le f_1(x) \Rightarrow OPT(y) \ge f_2(y)$
  - $OPT(x) > a(|x|)\,f_1(x) \Rightarrow OPT(y) < b(|y|)\,f_2(y)$

Figuring results



The PCP theorem

The class *NP*

Suppose that there is a verifier checking incoming proofs for prospective strings.

If $x \, \Xi \, L$ then there exists a proof that makes verifier accept.

If $x \, \mathrm{O} \, L$ then no proof makes verifier accept.

The class *PCP(logn,1)*

The verifier can read *O(logn)* random bits but only *O(1)* bits from the proof.

If $x \, \Xi \, L$ then there exists a proof that forces verifier to accept with probability *1*.

If $x \, \mathrm{O} \, L$ then on every proof verifier accepts with probability *<1/2*.

The PCP theorem

1. $PCP(\log n,1)\ \mathrm{N}\ NP$

Proof:

Simulate the PCP verifier for each random string of length *O(logn).*

These are polynomially many.

Accept if and only if all simulations accept, otherwise reject.

2. $NP\ \mathrm{N}\ PCP(\log n)$

Proof:

difficult (and omitted!)

<u>PCP theorem</u>: $NP = PCP(\log n,1).$

Hardness of MAX-3SAT

Max *k-function SAT*:

- n variables
- m functions of k of the n variables

Find truth assignment that maximizes the number of satisfied functions

Lemma: There is a constant k for which there is a gap-introducing reduction from *SAT* to *max k-function SAT* transforming φ to an instance I such that

- If φ is satisfiable, $OPT(I)=m$
- If φ is not satisfiable, $OPT(I)<(1/2)m$.

Hardness of MAX-3SAT

Proof:

Let V be a *PCP(logn,1)* verifier for *SAT*.

For each random string r of length $c\log n$, V reads q bits of the proof (a total of at most qn^c bits).

Introduce one variable for each of these bits

For fixed φ and r the verifier's answer depends only on the q bits that will read on the proof tape

For each r (and fixed φ) we introduce a function f_r which is a function of q variables (there are n^c such functions).

- If φ is satisfiable there is a proof that makes verifier accept with probability 1 and thus for all the random strings r, f_r is satisfied.
- If φ is not satisfiable then the acceptance probability is $<1/2$ which means that $<1/2$ of the random strings lead to acceptance. So $<1/2$ of the functions are satisfied.

Hardness of MAX-3SAT

Theorem:

There is a constant $\varepsilon_M > 0$ for which there is a gap-introducing reduction from *SAT* to *max-3SAT* that transforms a boolean formula φ to ψ such that:

- If φ is satisfiable, $OPT(\psi)=m$
- If φ is not satisfiable, $OPT(\psi)<(1-\varepsilon_M)m$.

Proof:

Using previous lemma we can transform the *SAT* formula to an instance of *max k-function SAT*.

Each f_r can be written as a SAT formula ψ_r

ψ is the conjuction of these ψ_r's.

- If φ is satisfiable then there is a proof that satisfies all the clauses of each ψ_r
- If φ is not satisfiable then for every proof every ψ_r must have one clause unsatisfied and so $>(1/2)n^c$ clauses of ψ unsatisfied.

MAX-3SAT with bounded occurrence

Theorem

There is a gap-preserving reduction from *MAX-3SAT* to *MAX-3SAT(29)* that transforms φ to ψ such that

- If $OPT(\varphi)=m$, then $OPT(\psi)=m'$
- If $OPT(\varphi)<(1-\varepsilon_M)m$, then $OPT(\psi)=(1-\varepsilon_b)m'$

Proof

For each variable $\underline{x}$ of $\underline{\varphi}$ that occurs $\underline{k}$ times we introduce a new set of k variables $x_1,\ldots,x_k$ and substitute each occurrence of x with one of these variables.

Additionally we construct a *14*-regular expander G on k vertices. We add to the formula the clauses $(x_i \text{ Ï } \overline{x}_j)$ and $(\overline{x}_i \text{ Ï } x_j)$ for each edge (x_i,x_j) of G.

We do this for all the variables of the formula and the resulting formula is ψ.

<u>Every optimal assignment for ψ must assign the same value to "same" variables</u>

- If φ is satisfiable so is ψ
- $OPT(\varphi)<(1-\varepsilon_M)m$ implies $>\varepsilon_M m$ clauses unsatisfied. Using the underlined remark ψ has $>\varepsilon_M m$ clauses unsatisfied.

Hardness of Vertex Cover

Theorem

There is a gap-preserving reduction from *max 3SAT(29)* to *VC(30)* that transforms a boolean formula φ to a graph $G=(V,E)$ such that

- If $OPT(\varphi)=m$, then $OPT(G) \le \frac{2}{3}|V|$
- If $OPT(\varphi)<(1-\varepsilon_b)m$, then $OPT(G) > (1+\varepsilon_u)\frac{2}{3}|V|$

Proof

The same reduction for showing NP-completenes. G has *3m* vertices.

Maximum independent set=OPT(φ):

- For an optimal truth assignment pick for each satisfied clause a literal that is satisfied. The corresponding vertices form an independent set
- For a maximum independent set *I* satisfy the coresponding literals. The "extension" of this assignment satisfies at least |*I*| clauses.

The complement of a max independent set is a minimum vertex cover

- If $OPT(\varphi)=m$ then $OPT(G)=2m$
- If $OPT(\varphi)<(1-\varepsilon_b)m$, then $OPT(G) > (2+\varepsilon_b)m = (1+\frac{\varepsilon_b}{2})\frac{2}{3}|V|$

Hardness of Steiner tree

Theorem

There is a gap-preserving reduction from *VC(30)* to the Steiner tree problem that transforms an instance of *G* of *VC(30)* to an instance *H=(R,S,cost)* satisfying:

$$OPT(G) \le \frac{2}{3}|V| \Rightarrow OPT(H) \le |R| + \frac{2}{3}|S| - 1$$

$$OPT(G) > (1+e_u)\frac{2}{3}|V| \Rightarrow OPT(H) > (1+e_s)(|R| + \frac{2}{3}|S| - 1)$$

Proof

Vertices of *H*

- Required: r_e, one for each edge of *G*
- Steiner: s_u, one for each vertex of *G*

Edge costs of H

- between *Steiner* vertices *cost=1*
- between *Required* vertices *cost=2*
- between *Required* vertex and “incident” *Steiner* vertex *cost=1*
- between all other pairs *cost=2*

$$VC(G) = c \Leftrightarrow ST(H) = |R| + c - 1$$

Hardness of Steiner tree

Proof (ctd.)

- For a vertex cover of size *c* let S_c be the corresponding *Steiner* vertices of *H*.
  H has a *steiner tree* with all edges of *cost 1* since each edge is incident to one vertex in *G*.
  It's total cost is $|R|+|S|-1=|R|+c-1$

- Let *T* be a Steiner tree of cost $|R|+c-1$.
  Let (*u,v*) be an edge of *cost 2* in *T*
  - Suppose *u* is *Steiner*. Remove (*u,v*) and add an edge from *v* to a *Required* vertex to connect the components. So both *u, v* "become" *Required*
  - Let e_u and e_v be the corresponding edges in *G*.
    G is connected so there is a path that includes both of them.
    Remove (*u,v*) disconnecting the tree.
    From the path there is a *Steiner* vertex that is connected to both the connected components
    Throw in these edges .

  T is transformed to have all edges with unit cost having the same total cost.
  Thus it has exactly *c Steiner* vertices. Their corresponding vertices in *G* form the required vertex cover of size *c*

Hardness of Clique

Lemma

There is a gap introducing reduction from SAT to Clique transforming φ of size n to a graph G of $2^q n^b$ vertices such that

- If φ is satisfiable, $OPT(G) \geq n^b$
- If φ is not satisfiable, $OPT(G) < \frac{1}{2} n^b$

Proof

Let F be a *PCP(logn,1)* verifier for *SAT*.

For each choice of r (length *blogn*), and each truth assignment τ, to q variables we get a vertex, say $u_{r,\tau}$. (total of $2^q n^b$ vertices)

We connect vertices that

- have an r so that if it "leads" to τ, then verifier accepts.
- their τ may be part of the same proof.

If φ is satisfiable let $p(r)$ be the part of the (good) proof that r "points". A clique of size n^b:

$$\{u_{r,p(r)} \mid r \text{ possible random choice}\}$$

If φ is not satisfiable then for every proof probability of acceptance is $<1/2$. So $<(1/2)n^b$ random choices "lead" to acceptance and so |*largest clique*| $<(1/2)n^b$.

(If we have a clique C then there is a proof for all τ of the "accepting" vertices. By this proof at least $|C|$ random choices lead to acceptance. Thus probability at least $|C|/n^b$.)

Generalizing the Verifier

We want something better.

Idea: Why don't we run the verifier more than once to obtain better results. We will need more random bits and read more bits!

The class *PCPc,s[r(s),q(n)]*

- The verifier can read *O(r(s))* random bits and *O(q(n))* bits from the proof.
- If $x \,\Xi\, L$ then there exists a proof, forcing verifier to accept with probability^{3} c.
- If $x \,\mathrm{O}\, L$ then on every proof verifier accepts with probability $< s$.

Ok. Simulate k times the verifier:

- reduce soundness (the s) to $<1/2^k$
- but O(klogn) random bits and
- O(k) bits querried

$NP=PCP_{1,1/n}[\log n, \log n]$

Proof:

Let $L \, \Xi \, PCP(\log n, 1)$ decided by verifier *F.* If we simulate the verifier *O(logn)* times we have *O(logn)* bits querried but we will need $O(\log^2 n)$ random bits.

Use expanders.

- Construct an expander with n^b vertices labeled with *O(blogn)* bits.
- Pick a vertex at random and take a random walk of length *O(logn).*
- Simulate the verifier *O(logn)* times using as random bits the labels of the vertices of the path.
- Accept iff all simulations accept

If $x \, \Xi \, L$ then all simulations will accept

If $x \, \mathrm{O} \, L$ then F accepts for $<n^b/2$ random strings. Expanders ensure us that the probability that the path has only “accepting” vertices is $<1/n$.

(New) Hardness of Clique

Lemma

There is a gap introducing reduction from SAT to Clique transforming φ of size n to a graph G of n^{b+q} vertices such that

- If φ is satisfiable, $OPT(G) \geq n^{b}$
- If φ is not satisfiable, $OPT(G) < n^{b-1}$

Proof

Let F be a $PCP_{1,1/n}(logn,logn)$ verifier for *SAT*.

For each choice of r (length *blogn*), and each truth assignment τ, to *qlogn* variables we get a vertex, say $u_{r,\tau}$. (total of n^{b+q} vertices)

We connect vertices that

- have an r that "leads" to τ that "leads" to acceptance.
- their τ may be part of the same proof.

If φ is satisfiable let $p(r)$ be the part of the proof that r "points". A clique of size n^{b}

$\{u_{r,p(r)} \mid r \text{ possible random choice}\}$

If φ is not satisfiable then for every proof probability of acceptance is $<1/n$. So $<n^{b-1}$ random choices "lead" to acceptance and so $|\text{largest clique}| < n^{b-1}$.

Another characterization for NP

The *two prover one round* model

- There are two proofs (provers, non communicating)
- *O(logn)* random bits can be used by the verifier and
- One position of each proof can be queried

The class *2P1R$_{c,s}$(r(n))*

$L \,\Xi\, 2P1R_{c,s}(\mathrm{r(n)})$ if there is a p.t. verifier that reads *O(r(n))* random bits and for every input x

- If $x \,\Xi\, L$, there is a pair of proofs that makes verifier to accept with probability ³ c
- If $x \,\mathrm{O}\, L$, for every pair of proofs verifier accept with probability $< s$

Theorem

NP=2P1R$_{1,1\text{-}e}$(log(n)) (for some constant e>0)

$NP \text{ N } 2P1R_{1,1-e}(\log n)$

Proof

We can map a boolean formula φ to an instance ψ of *Max3Sat(5)* so that

- If φ is satisfiable, $OPT(\psi)=m$
- If φ is not satisfiable, $OPT(\psi)<(1-\varepsilon)m$

The verifier:

- does the above reduction
- “gets”
  - a proof containing a truth assignment for ψ and
  - another containing in each position the truth assignments for each clause (encoded)
- uses $O(logn)$ random bits to pick
  - a clause C
  - a variable x of the clause
- asks
  - first proof for the value of x
  - second proof for the values of the variables of C (including x)
- accepts iff C is satisfied and the two assignments of x agree.

- If φ is satisfiable so is ψ and so there is a pair of proofs forcing verifier to accept
- If φ is not satisfiable then suppose τ,z the two proofs
  - at least εm clauses unsatisfied by assignment τ.
  - C is unsatisfied with probability $>\varepsilon$ (under τ).
  - if that is the case and z satisfies C then τ,z disagree at least at one assignment on the variables of C.
  - V catches this with probability $\varepsilon/3$.