Hardness of Approximation

Olli Pottonen
olli.pottonen@tkk.fi

April 28, 2008

Reductions and gaps

- *Gap-introducing* and *gap-preserving* reductions
- Gap-introducing reduction from SAT to minimization problem Π: $\phi \mapsto x$ and
  - if ϕ is satisfiable, $OPT(x) \leq f(x)$
  - if ϕ is not satisfiable, $OPT(x) > \alpha(|x|)f(x)$
- Gap-preserving reductiong from minimazition problem Π_1 to minimization problem Π_2: $x_1 \mapsto x_2$ and
  - $OPT(x_1) \leq f_1(x_1) \Rightarrow OPT(x_2) \leq f_2(x_2)$
  - $OPT(x_1) > \alpha(|x_1|)f_1(x_1) \Rightarrow OPT(x_2) > \beta(|x_2|)f_2(x_2)$

Probabilistically checkable proof (PCP)

- First step in gap-introducing reductions
- Recall the definition of NP: language L is in NP if there is a deterministic polynomial time verifier V such that
  - if $x \in L$, then there is a polynomial-sized proof that makes V accept
  - if $x \notin L$, then no proof makes V accept

PCP(f, g)

- Language L is in $PCP(f, g)$, if there is a probabilistic verifier V which takes $O(f)$ bits of randomness and inspects $O(g)$ bits of the proof such that
  - if $x \in L$, then there is a polynomial-sized proof that makes V accept with probability 1
  - if $x \notin L$, then no proof makes V accept with probability $\geq 1/2$

The PCP theorem

- $PCP(\log n, 1) = NP$
- $PCP(\log n, 1) \subseteq NP$, since all $2^{O(\log n)} = n^c$ possible computations can be checked in polynomial time
- The difficult part: $NP \subseteq PCP(\log n, 1)$. Proof omitted.

What does this have to do with approximation?

- **Maximize accept probability** Consider a $PCP(\log n, 1)$ verifier for SAT. For SAT formula ϕ, find the proof which maximizes acceptance probability.

- By the PCP theorem, no factor $1/2$ approximation algorithm unless $P = NP$.

Next goal: MAX-3SAT

- We wish to construct gap-introducing reduction from SAT to MAX-3SAT that transforms a Boolean formula ϕ to ψ with m clauses such that
  - ϕ satisfiable $\Rightarrow OPT(\psi) = m$, and
  - ϕ not satisfiable $\Rightarrow OPT(\psi) < (1 - \epsilon_M)m$

  for some constant $\epsilon_M > 0$.

MAX k-FUNCTION SAT

- **MAX k-FUNCTION SAT** Given n Boolean functions on m variables such that each functions takes a constant number k of arguments, maximize the number of satisfied functions.

- For some constant k there is a gap-introducing reduction from SAT to MAX k-function SAT that transforms a formula ϕ to an instance I with m functions such that
  - ϕ satisfiable $\Rightarrow OPT(I) = m$, and
  - ϕ not satisfiable $\Rightarrow OPT(I) < \frac{1}{2}m$.

- Proof: consider $PCP(\log n, 1)$ verifier and take one function for each possible computation.

MAX-3SAT

- Given MAX k-FUNCTION SAT instance I, transform each function to a 3SAT formula J. Assume we originally have n^c functions. The transform results in $m \leq n^c 2^k (k-2)$ clauses. If $OPT(I) = n^c$, then $OPT(J) = m$, if $OPT(I) < \frac{1}{2} n^c$, ther $OPT(J) \leq 1/2\epsilon n^c$ with $\epsilon = 1/(2^k(k-2))$.

Clique

- For some positive ϵ, there is no $1/n^{\epsilon}$ factor approximation algorithm unless $P = NP$.

- First let us proof that there is no factor $1/2$ approximation algorithm:

- For constants b, Q, there is a gap-introducing reduction from SAT to clique which transforms a formula ϕ of size n to a graph G with $|V| = Qn^b$ such that
  - if ϕ is satisfiable, then $OPT(G) \geq n^b$
  - if ϕ is not satisfiable, then $OPT(G) < \frac{1}{2}n^b$

Reduction from SAT to clique

- Consider $PCP(\log n, 1)$ verifier for F SAT. It requires $b \log n$ bits of randomness and q bits of proof. For each possible computation of F, construct a vertex $v_{r,\tau}$, where r and τ are the random bits and proof bits read by F, respectively. There are $2^q n^b = Qn^b$ vertices. Vertices are adjacent if they are accepting and have non-contradicting proof bits. If there is a clique of size k, there is at least one proof consistent with the clique. The proof is accepted with probability at least $k/(Qn^b)$.

Towards better reduction

- If the verifier accepts false proof with probability $< 1/$, then the same reduction would have gap size $1/n^{\epsilon}$ instead of $1/2$.

- $PCP_{c,s}(f, g)$: correct proof accepted with probability c, false with probability s. We would like to have $s = 1/n$ instead of $1/2$.

- Standard trick: repeat computation $O(\log n)$ times to get error probability $1/n$.

- Problem: $O(\log n)$ runs with $O(\log n)$ random bits each time requires $O(\log^2 n)$ random bits. This is too much!

- Solution: first take random string r of length $b \log n$, make small changes $O(\log n)$ times, each change requiring $O(1)$ bits of randomness. Now we get $O(\log n)$ random strings.

- Random walk in an expander graph.

- Theorem: Assume constant degree expander graph H with n^b vertices. There is a constant k such that for any set S of vertices with size $< n^b/2$, $Pr(\text{random walk of length } k \log n \text{ lies in } S) < 1/n$.

- Thus: $NP = PCP(\log n, 1) = PCP_{1,1/n}(\log n, \log n)$.