

Today

PCP Theorem, Simple proof due to Irit Dinur [ECCC, TR05-046]!

Few Comments

- Based on some ideas promoted in [Dinur-Reingold '04].
- Remarkably simple novel proof.
- Leads to new quantitative results too!
- Invites others to work on PCPs.

Outline

- Define approximation of MAX SAT.
- Equivalence of approximating MAX SAT and PCP constructions.
- Prove theorem in terms of MAX SAT.

Approximations

Example optimization problem: Given a graph, color it with as few colors as possible.

We know this is NP-hard (since coloring 3-colorable graphs with 3 colors is hard).

But can you always color such graphs with 4 colors?

Can you always color k -colorable graphs with $k+1$ colors?

Approximability: Study of finding such “near optimal” solutions.

$\alpha(\cdot)$ -approx. algorithm (runs in poly time) and produces a solution whose cost $\leq \alpha(n) \cdot \text{OPT}$ (or profit $\geq \text{OPT}/\alpha(n)$).

Format for hardness reduction

Example: Algorithm that computes a 4-coloring of every planar graph is a $4/3$ -approx. algorithm for coloring planar graphs. . What about the general case: Best known algorithm colors 3-colorable graph with $n^{3/14}$ colors.

Is this best possible? If so, what would be the "nature" of a theorem that proves this?

There exists a transformation T mapping 3cnf formulae to graphs such that if $\phi \in \text{SAT}$ then $T(\phi)$ is 3-colorable, and if $\phi \notin \text{SAT}$ then $T(\phi)$ is not $n^{3/14}$ -colorable.

(Aside: best known such transformation only guarantees " $T(\phi)$ is not 4-colorable" in latter case. :-())

MAX SAT Problems

Max k -SAT- Σ :

Instance: Constraints C_1, \dots, C_m on variables X_1, \dots, X_n , where variables take on values in Σ and constraints are arbitrary Boolean functions on upto k variables each.

Goal: Find assignment to variables maximizing number of satisfied constraints.

Aside: In general not all that interesting, though special cases are interesting. More importantly very useful in showing hardness of other problems (such as coloring etc.)

Exercise: Show that if MAX k -SAT- Σ is hard

to approximate to within $\alpha > 1$, then there exists $\alpha' > 1$ such that MAX k' -SAT- Σ' is hard to approximate to within α' provided $k, |\Sigma'| \geq 2$.

PCP and MAX SAT

Claim 1: If MAX k -SAT- Σ is hard to approximate within α (by reduction in above format with good case being satisfiable), then $\text{NP} \subseteq \text{PCP}[O(\log n), O(1)]$.

Proof: Verifier transforms NP problem to Max SAT instance and expects as proof an assignment to the variables. Verifies proof by picking a random constraint and verifying it is satisfied. Query complexity is $k \log |\Sigma|$, and accepts invalid theorems w.p. at most $1/\alpha$.

Claim 2: If $\text{NP} \subseteq \text{PCP}[O(\log n), q]$, then MAX q -SAT- $\{0, 1\}$ is hard to approximate.

Proof: Let X_1, \dots, X_n denote the bits of the proof. Let $m = 2^{O(\log n)}$ denote the number

of random strings available to the verifier. C_j applies the constraint saying that the verifier accepts on the j th random string. Note that this constraint depends on only q bits (2^q if verifier is adaptive).

Dinur's Theorem

Given instance ψ of MAX k -SAT- Σ , and assignment σ to the variable of ψ , let $\text{unsat}_\sigma(\psi)$ denote the fraction of clauses left unsatisfied by σ . Let $\text{unsat}(\psi) = \min_\sigma \{\text{unsat}_\sigma(\psi)\}$.

Theorem: There exists a transformation T' transforming 3cnf formulae to MAX k -SAT- Σ such that if $\phi \in \text{SAT}$ then $T'(\phi) \in \text{SAT}$ and if $\phi \notin \text{SAT}$ then $\text{unsat}(T'(\phi)) > \epsilon$.

Main Lemma: For some constant Σ , $\epsilon > 0$, there exists a transform T transforming MAX 2-SAT- Σ instances to MAX 2-SAT- Σ preserving satisfiability such that $\text{unsat}(T(\phi)) \geq \min\{\epsilon, 2 \cdot \text{unsat}(\phi)\}$. Furthermore size of $T(\phi)$ is $O(|\phi|)$.

Theorem from Main Lemma

- Step 1: Transform T_0 the given 3cnf formula in instance of MAX 2-SAT- Σ such that $T_0(\phi)$ is satisfiable iff ϕ is satisfiable. Note $\text{unsat}(T_0(\phi)) \geq 1/m = 1/\text{poly}(n)$.
- Step 2: Apply the transform T to $T_0(\phi)$ $\log_2 m$ times.
- Analysis:
 - The resulting formula, denoted $T'(\phi)$ has size $C^{\log m} \cdot |\phi| = \text{poly}(n) \cdot |\phi|$.
 - If ϕ is satisfiable, so is $T'(\phi)$.
 - If ϕ is not satisfiable, then $\text{unsat}(T'(\phi)) \geq \min\{\epsilon, 2^{\log_2 m} \text{unsat}(T_0(\phi))\} = \epsilon$.

Proof of Main Lemma

also alphabet size. Is it a good idea to combine? Yes ... courtesy of quantifiers.

Two substeps:

Lemma 1 (Gap Amplification): For every β , Σ , there exists a ℓ and a transform T_1 from MAX 2-SAT- Σ to MAX 2-SAT- Σ^ℓ preserving satisfiability such that $\text{unsat}(T_1(\phi)) \geq \beta \cdot \text{unsat}(\phi)$. Furthermore $|T_1(\phi)| = O(|\phi|)$.

Lemma 2 (Alphabet reduction): There exists Σ and constant c such that for every finite Γ , there is a transform T_2 from MAX 2-SAT- Γ to MAX 2-SAT- Σ preserving satisfiability such that $\text{unsat}(T_2(\phi)) \geq \frac{1}{c} \cdot \text{unsat}(\phi)$. Furthermore $|T_2(\phi)| = O(|\phi|)$.

Notes: Lemma 1 increases gap, but also alphabet size. Lemma 2 decreases gap but

Main Lemma from Sub-Lemmas

Alphabet Reduction Substeps

- Set Σ as in Lemma 2.
- Set $\beta = 2c$, and let $\Gamma = \Sigma^\ell$, where ℓ is from Lemma 1. Let $T(\phi) = T_2(T_1(\phi))$.
- Then T is linear sized, poly time and preserves satisfiability. Furthermore

$$\begin{aligned}
 \text{unsat}(T(\phi)) &= \text{unsat}(T_2(T_1(\phi))) \\
 &\geq \frac{1}{c} \cdot \text{unsat}(T_1(\phi)) \\
 &\geq \frac{\beta}{c} \cdot \text{unsat}(\phi) \\
 &= 2 \cdot \text{unsat}(\phi)
 \end{aligned}$$

Lemma 2a: There exists a constant k , and c' such that for every finite Γ , there is a transform T_{2a} from MAX 2-SAT- Γ to MAX k -SAT- $\{0, 1\}$ preserving satisfiability such that $\text{unsat}(T_{2a}(\phi)) \geq \frac{1}{c'} \cdot \text{unsat}(\phi)$. Furthermore T_{2a} is linear.

Lemma 2b: For every k , there is a linear sized, satisfiability preserving transform T_{2b} transforming MAX k -SAT- $\{0, 1\}$ to MAX 2-SAT- $\{0, 1\}^k$ such that $\text{unsat}(T_{2b}(\phi)) \geq \frac{1}{k} \cdot \text{unsat}(\phi)$.

Lemma 2 follows with $\Sigma = 2^k$ and $c = c' \cdot k$, for $T_2 = T_{2b} \circ T_{2a}$.

Based on reduction from oracle interactive proofs to 2-prover interactive proofs [FRS].

- Given instance ϕ with X_1, \dots, X_n and constraints C_1, \dots, C_m , create instance ψ with variables X'_1, \dots, X'_n and Y_1, \dots, Y_m and constraints C'_{11}, \dots, C'_{km} with C'_{ij} being the following:
 - Suppose C_j applies predicate P to variables $X_{v_1(j)}, \dots, X_{v_k(j)}$.
 - Then C'_{ij} accepts if $f(Y_j) = 1$ and $X'_{v_i(j)} \in \{0, 1\}$ and $X'_{v_i(j)} = (Y_j)_i$ (where $(Y_j)_i$ denotes i th bit of $Y_j \in \{0, 1\}^k$).
- Analysis left as easy exercise.

(First hard lemma (to prove))

- Need a “gadget”: Takes variables $X_i \in \Gamma$ to $X'_{i,1}, \dots, X'_{i,t} \in \Sigma$. For constraint C_j , adds variables $Y_{j,1}, \dots, Y_{j,t'}$, and new constraints $C'_{j,1}, \dots, C'_{j,s}$ along with maps $E : \Gamma \rightarrow \Sigma^t$ and $D : \Sigma^t \rightarrow \Gamma$ such that
 - If C_j on (X_{i1}, X_{i2}) is satisfied, then assigning $E(X_{i1})$ to $X'_{i1,1}, \dots, X'_{i1,t}$ and $E(X_{i2})$ to $X'_{i2,1}, \dots, X'_{i2,t}$, leads to constraints $C'_{j,1}, \dots, C'_{j,s}$ that are satisfiable by appropriate choice of $Y_{j,1}, \dots, Y_{j,t'}$.
 - If some assignment to $X'_{i1,1}, \dots, X'_{i1,t}; X'_{i2,1}, \dots, X'_{i2,t};$ and $Y_{j,1}, \dots, Y_{j,s}$

satisfies $1 - \frac{1}{c}$ fraction of constraints $C'_{j,1}, \dots, C'_{j,s}$ then setting $X_i = D(X'_{i,1}, \dots, X'_{i,t})$ satisfies C_j .

- “Gadgets” as above common in Garey-Johnson. Unfortunately typically have $\Gamma = \Sigma^k$ and $c = k$. We can’t afford this. How to get a better gadget?
- Insight [Arora-Safra]: PCP theorems typically produce such gadgets. E.g., can get one from our exponential sized PCP for NP. Will do even better by considering a doubly exponential sized PCP for NP.