PCPs and Hardness of Approximation

Lydia Zakynthinou

MPLA, NTUA

February 2, 2015

Outline

Outline

Approximation Algorithms

Desired Algorithm Properties:

- For every input...
- runs in polynomial time
- returns the optimal solution

Approximation Algorithms

Desired Algorithm Properties:

- For every input...
- runs in polynomial time
- returns the optimal solution

Approximation Algorithms

Desired Algorithm Properties:

- For every input...
- runs in polynomial time
- returns the optimal solution

Approximation Algorithms

Desired Algorithm Properties:

- For every input...
- runs in polynomial time
- returns the optimal solution

Approximation Algorithms

Approximation algorithm:

- For every input...
- runs in polynomial time
- returns an approximate solution.

Definition

A ρ-approximate solution S for a maximization (respectively, minimization) problem with cost function V and $\rho \geq 1$ is a feasible solution with $V(S) \geq \frac{1}{\rho} V(OPT)$ (respectively, $V(S) \leq \rho V(OPT)$).

Definition

A ρ-approximate algorithm returns a ρ-approximate solution for every input.

Approximation Algorithms

Approximation algorithm:

- For every input...
- runs in polynomial time
- returns an approximate solution.

Definition

A ρ-approximate solution S for a maximization (respectively, minimization) problem with cost function V and $\rho \geq 1$ is a feasible solution with $V(S) \geq \frac{1}{\rho} V(OPT)$ (respectively, $V(S) \leq \rho V(OPT)$).

Definition

A ρ-approximate algorithm returns a ρ-approximate solution for every input.

Approximation Algorithms

Approximation algorithm:

- For every input...
- runs in polynomial time
- returns an approximate solution.

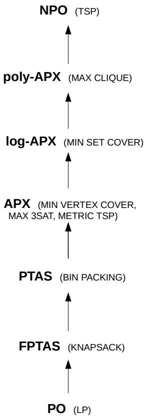
Definition

A ρ-approximate solution S for a maximization (respectively, minimization) problem with cost function V and $\rho \geq 1$ is a feasible solution with $V(S) \geq \frac{1}{\rho}V(OPT)$ (respectively, $V(S) \leq \rho V(OPT)$).

Definition

A ρ-approximate algorithm returns a ρ-approximate solution for every input.

Classes of Optimization Problems



Hardness of Approximation

Using classic $\mathbf{NP}$-complete reductions, we sometimes get inapproximability results:

Theorem

There is no constant factor approximation for TSP, unless $\mathbf{P} = \mathbf{NP}$.

In general, we need a new type of reduction:

Theorem

Let a language L and w.l.o.g. a maximization problem P. If the following holds:

- $x \in L \Rightarrow V(S_{R(x)}^{OPT}) \geq c$
- $x \notin L \Rightarrow V(S_{R(x)}^{OPT}) < s$

then: if there is a polynomial time c/s-approximate algorithm for P and L is $\mathbf{NP}$*-complete, then* $\mathbf{P} = \mathbf{NP}$.

Hardness of Approximation

Using classic $\mathbf{NP}$-complete reductions, we sometimes get inapproximability results:

Theorem

There is no constant factor approximation for TSP, unless $\mathbf{P} = \mathbf{NP}$.

In general, we need a new type of reduction:

Theorem

Let a language L and w.l.o.g. a maximization problem P. If the following holds:

- $x \in L \Rightarrow V(S_{R(x)}^{OPT}) \geq c$
- $x \notin L \Rightarrow V(S_{R(x)}^{OPT}) < s$

then: if there is a polynomial time c/s-approximate algorithm for P and L is $\mathbf{NP}$*-complete, then* $\mathbf{P} = \mathbf{NP}$.

Hardness of Approximation

Using classic $\mathbf{NP}$-complete reductions, we sometimes get inapproximability results:

Theorem

There is no constant factor approximation for TSP, unless $\mathbf{P} = \mathbf{NP}$.

In general, we need a new type of reduction:

Theorem

Let a language L and w.l.o.g. a maximization problem P. If the following holds:

- $x \in L \Rightarrow V(S^{OPT}_{R(x)}) \geq c$
- $x \notin L \Rightarrow V(S^{OPT}_{R(x)}) < s$

then: if there is a polynomial time c/s-approximate algorithm for P and L is $\mathbf{NP}$*-complete, then* $\mathbf{P} = \mathbf{NP}$.

Hardness of Approximation

Using classic $\mathbf{NP}$-complete reductions, we sometimes get inapproximability results:

Theorem

There is no constant factor approximation for TSP, unless $\mathbf{P} = \mathbf{NP}$.

In general, we need a new type of reduction:

Theorem

Let a language L and w.l.o.g. a maximization problem P. If the following holds:

- $x \in L \Rightarrow V(S_{R(x)}^{OPT}) \geq c$
- $x \notin L \Rightarrow V(S_{R(x)}^{OPT}) < s$

then: if there is a polynomial time c/s-approximate algorithm for P and L is $\mathbf{NP}$-complete, then $\mathbf{P} = \mathbf{NP}$.

Hardness of Approximation

Using classic $\mathbf{NP}$-complete reductions, we sometimes get inapproximability results:

Theorem

There is no constant factor approximation for TSP, unless $\mathbf{P} = \mathbf{NP}$.

In general, we need a new type of reduction:

Theorem

Let a language L and w.l.o.g. a maximization problem P. If the following holds:

- $x \in L \Rightarrow V(S_{R(x)}^{OPT}) \geq c$
- $x \notin L \Rightarrow V(S_{R(x)}^{OPT}) < s$

then: if there is a polynomial time c/s-approximate algorithm for P and L is $\mathbf{NP}$*-complete, then* $\mathbf{P} = \mathbf{NP}$.

Hardness of Approximation

Using classic $\mathbf{NP}$-complete reductions, we sometimes get inapproximability results:

Theorem

There is no constant factor approximation for TSP, unless $\mathbf{P} = \mathbf{NP}$.

In general, we need a new type of reduction:

Theorem

Let a language L and w.l.o.g. a maximization problem P. If the following holds:

- $x \in L \Rightarrow V(S^{OPT}_{R(x)}) \geq c$
- $x \notin L \Rightarrow V(S^{OPT}_{R(x)}) < s$

then: if there is a polynomial time c/s-approximate algorithm for P and L is $\mathbf{NP}$*-complete, then* $\mathbf{P} = \mathbf{NP}$.

The PCP Theorem: Characterization of $\mathbf{NP}$

Theorem (Arora, Lund, Motwani, Sudan, Szegedy '92)

$\mathbf{NP} = \mathbf{PCP}[\mathrm{O}(\log n), \mathrm{O}(1)]$

Outline

MAX-3SAT Inapproximability

Theorem

The PCP Theorem implies that there is an ϵ_1 such that there is no polynomial time $(1+\epsilon_1)$-approximation algorithm for MAX-3SAT, unless $\mathbf{P}=\mathbf{NP}$.

Let $L \in \mathbf{PCP}[O(\log n), q]$ be an $\mathbf{NP}$-complete language.

We construct a 3CNF formula such that for some $\epsilon_1 > 0$:

- $x \in L \Rightarrow \phi_x$ is satisfiable
- $x \notin L \Rightarrow$ no assignment satisfies more than $(1-\epsilon_1)m$ clauses of ϕ_x

MAX-3SAT Inapproximability

Theorem

The PCP Theorem implies that there is an ϵ_1 such that there is no polynomial time $(1 + \epsilon_1)$-approximation algorithm for MAX-3SAT, unless $\mathbf{P} = \mathbf{NP}$.

Let $L \in \mathbf{PCP}[\mathrm{O}(\log n), q]$ be an $\mathbf{NP}$-complete language.

We construct a 3CNF formula such that for some $\epsilon_1 > 0$:

- $x \in L \Rightarrow \phi_x$ is satisfiable
- $x \notin L \Rightarrow$ no assignment satisfies more than $(1 - \epsilon_1)m$ clauses of ϕ_x

MAX-3SAT Inapproximability

Theorem

The PCP Theorem implies that there is an ϵ_1 such that there is no polynomial time $(1+\epsilon_1)$-approximation algorithm for MAX-3SAT, unless $\mathbf{P} = \mathbf{NP}$.

Let $L \in \mathbf{PCP}[O(\log n), q]$ be an $\mathbf{NP}$-complete language.

We construct a 3CNF formula such that for some $\epsilon_1 > 0$:

- $x \in L \Rightarrow \phi_x$ is satisfiable
- $x \notin L \Rightarrow$ no assignment satisfies more than $(1-\epsilon_1)m$ clauses of ϕ_x

MAX-3SAT Inapproximability

For every random string $r \in \{0,1\}^{O(\log n)}$, the Verifier checks q bits $i_r^1, \ldots, i_r^q$ of the proof y and accepts iff $f_r^x(y_{i_r^1}, \ldots, y_{i_r^q}) = \mathbf{true}$.

f_r^x can be simulated by a qCNF formula of at most 2^q clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,2^q}^x$

...which is a 3CNF formula of at most $(q-2)2^q$ clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,(q-2)2^q}^x$

MAX-3SAT Inapproximability

For every random string $r \in \{0,1\}^{O\left(\log n\right)}$, the Verifier checks q bits $i_r^1, \ldots, i_r^q$ of the proof y and accepts iff $f_r^x(y_{i_r^1}, \ldots, y_{i_r^q}) = \mathbf{true}$.

f_r^x can be simulated by a qCNF formula of at most 2^q clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,2^q}^x$

...which is a 3CNF formula of at most $(q-2)2^q$ clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,(q-2)2^q}^x$

MAX-3SAT Inapproximability

For every random string $r \in \{0,1\}^{O(\log n)}$, the Verifier checks q bits $i_r^1, \ldots, i_r^q$ of the proof y and accepts iff $f_r^x(y_{i_r^1}, \ldots, y_{i_r^q}) = \mathbf{true}$.

f_r^x can be simulated by a qCNF formula of at most 2^q clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,2^q}^x$

...which is a 3CNF formula of at most $(q-2)2^q$ clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,(q-2)2^q}^x$

MAX-3SAT Inapproximability

For every random string $r \in \{0,1\}^{O(\log n)}$, the Verifier checks q bits $i_r^1, \ldots, i_r^q$ of the proof y and accepts iff $f_r^x(y_{i_r^1}, \ldots, y_{i_r^q}) = \mathbf{true}$.

f_r^x can be simulated by a qCNF formula of at most 2^q clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,2^q}^x$

...which is a 3CNF formula of at most $(q-2)2^q$ clauses:
$f_r^x = C_{r,1}^x \wedge \ldots \wedge C_{r,(q-2)2^q}^x$

MAX-3SAT Inapproximability

How many "r"s are there? $\rightarrow 2^{O\left(\log n\right)} = 2^{k \log n} = n^k$ for some constant k.

- $x \in L \Rightarrow \mathbf{Pr}[\text{V accepts}] = 1 \Rightarrow f^x_{r_1} \wedge \ldots \wedge f^x_{r_{n^k}} = \mathbf{true}$
- $x \notin L \Rightarrow \mathbf{Pr}[\text{V accepts}] \leq 1/2 \Rightarrow \leq n^k(q-2)2^q - \frac{n^k}{2}$ clauses satisfied

$\rightarrow \epsilon_1 = \frac{1}{2 \cdot (q-2) \cdot 2^q}$

MAX-3SAT Inapproximability

How many "r"s are there? $\rightarrow 2^{O\left(\log n\right)} = 2^{k \log n} = n^k$ for some constant k.

- $x \in L \Rightarrow \mathbf{Pr}[\text{V accepts}] = 1 \Rightarrow f^x_{r_1} \wedge \ldots \wedge f^x_{r_{n^k}} = \mathbf{true}$
- $x \notin L \Rightarrow \mathbf{Pr}[\text{V accepts}] \leq 1/2 \Rightarrow \leq n^k(q-2)2^q - \frac{n^k}{2}$ clauses satisfied

$\rightarrow \epsilon_1 = \frac{1}{2 \cdot (q-2) \cdot 2^q}$

MAX-3SAT Inapproximability

How many "r"s are there? $\rightarrow 2^{O\left(\log n\right)} = 2^{k \log n} = n^k$ for some constant k.

- $x \in L \Rightarrow \mathbf{Pr}[\text{V accepts}] = 1 \Rightarrow f^x_{r_1} \wedge \ldots \wedge f^x_{r_{n^k}} = \mathbf{true}$
- $x \notin L \Rightarrow \mathbf{Pr}[\text{V accepts}] \leq 1/2 \Rightarrow \leq n^k(q-2)2^q - \frac{n^k}{2}$ clauses satisfied

$\rightarrow \epsilon_1 = \frac{1}{2 \cdot (q-2) \cdot 2^q}$

MAX-3SAT Inapproximability

How many "r"s are there? $\rightarrow 2^{O(\log n)} = 2^{k \log n} = n^k$ for some constant k.

- $x \in L \Rightarrow \mathbf{Pr}[\text{V accepts}] = 1 \Rightarrow f^x_{r_1} \wedge \ldots \wedge f^x_{r_{n^k}} = \mathbf{true}$
- $x \notin L \Rightarrow \mathbf{Pr}[\text{V accepts}] \leq 1/2 \Rightarrow \leq n^k(q-2)2^q - \frac{n^k}{2}$ clauses satisfied

$\rightarrow \epsilon_1 = \frac{1}{2 \cdot (q-2) \cdot 2^q}$

MAX-3SAT Inapproximability

How many "r"s are there? $\rightarrow 2^{O\left(\log n\right)} = 2^{k \log n} = n^k$ for some constant k.

- $x \in L \Rightarrow \mathbf{Pr}[\text{V accepts}] = 1 \Rightarrow f^x_{r_1} \wedge \ldots \wedge f^x_{r_{n^k}} = \mathbf{true}$
- $x \notin L \Rightarrow \mathbf{Pr}[\text{V accepts}] \leq 1/2 \Rightarrow \leq n^k(q-2)2^q - \frac{n^k}{2}$ clauses satisfied

$\rightarrow \epsilon_1 = \frac{1}{2\cdot(q-2)\cdot 2^q}$

MAX-3SAT Inapproximability

The converse holds:

Theorem

If there is a reduction like the previous one for an $\mathbf{NP}$-complete L, then the PCP Theorem holds.

Hint: The Verifier constructs ϕ_x, picks $O\left(\frac{1}{\epsilon_1}\right)$ clauses at random and accepts iff they are satisfied.

MAX-3SAT-d Inapproximability

Theorem

If there is an approximation algorithm for MAX-E3SAT-d with performance better than $(1 - \epsilon_2)$*, then* $\mathbf{P} = \mathbf{NP}$*.*

Similar reductions show that if $\mathbf{P} = \mathbf{NP}$, then there is no approximation for:

- MAX-IS in degree-$(d + 2)$ graphs with performance ratio better than $1/(1 - \epsilon_2)$

Let ϕ have n variables and m clauses. Construct a graph G_ϕ with one vertex for every occurence of a variable $(3m)$ and edges between the variables of a clause and their negations. There is an IS of size $\geq t$ iff there is an assignment satisfying $\geq t$ clauses.

MAX-3SAT-d Inapproximability

Theorem

If there is an approximation algorithm for MAX-E3SAT-d with performance better than $(1-\epsilon_2)$, then $\mathbf{P}=\mathbf{NP}$.

Similar reductions show that if $\mathbf{P}=\mathbf{NP}$, then there is no approximation for:

- MAX-IS in degree-$(d+2)$ graphs with performance ratio better than $1/(1-\epsilon_2)$

Let ϕ have n variables and m clauses. Construct a graph G_ϕ with one vertex for every occurence of a variable $(3m)$ and edges between the variables of a clause and their negations. There is an IS of size $\geq t$ iff there is an assignment satisfying $\geq t$ clauses.

MAX-3SAT-d Inapproximability

Theorem

If there is an approximation algorithm for MAX-E3SAT-d with performance better than $(1-\epsilon_2)$, then $\mathbf{P} = \mathbf{NP}$.

Similar reductions show that if $\mathbf{P} = \mathbf{NP}$, then there is no approximation for:

- MAX-IS in degree-$(d+2)$ graphs with performance ratio better than $1/(1-\epsilon_2)$

Let ϕ have n variables and m clauses. Construct a graph G_ϕ with one vertex for every occurence of a variable $(3m)$ and edges between the variables of a clause and their negations. There is an IS of size $\geq t$ iff there is an assignment satisfying $\geq t$ clauses.

MAX-3SAT-d Inapproximability

Similar reductions show that if $\mathbf{P} = \mathbf{NP}$, then there is no approximation for:

- MAX-IS in degree-$(d + 2)$ graphs with performance ratio better than $1/(1 - \epsilon_2)$
- MIN VERTEX COVER in degree-$(d + 2)$ graphs with performance ratio better than $1 + \epsilon_2/2$
- MIN METRIC STEINER TREE with performance ratio $1 + \epsilon_3$

MAX-3SAT-d Inapproximability

Similar reductions show that if $\mathbf{P} = \mathbf{NP}$, then there is no approximation for:

- MAX-IS in degree-$(d + 2)$ graphs with performance ratio better than $1/(1 - \epsilon_2)$
- MIN VERTEX COVER in degree-$(d + 2)$ graphs with performance ratio better than $1 + \epsilon_2/2$
- MIN METRIC STEINER TREE with performance ratio $1 + \epsilon_3$

MAX-3SAT-d Inapproximability

Similar reductions show that if $\mathbf{P} = \mathbf{NP}$, then there is no approximation for:

- MIN METRIC STEINER TREE with performance ratio $1 + \epsilon_3$

Let G be a graph in the MIN VERTEX COVER problem. We construct a new G' with one vertex for every vertex of G and one for every edge of G.

- For every edge (u, v) of G: $d_{G'}([u], [u, v]) = 1$.
- For every two vertices of G: $d_{G'}([u], [v]) = 1$.
- Every other vertex pair of G' has distance 2.

There is a vertex cover of k vertices in G iff there is a Steiner Tree of cost $\leq m - k$ in G'.

IS Inapproximability

Let $L \in \mathbf{PCP}_{c,s}[\mathrm{O}(\log n), \mathrm{O}(1)]$. Configuration is the computation of the Verifier for random bits r and queries q ($\rightarrow 2^q 2^{k \log n}$ configurations). Consider the graph of accepting configurations G_x, where there is an edge between every inconsistent configurations.

- $x \in L \Rightarrow$ there is an IS in G_x of size $\geq c \cdot 2^{k \log n}$
- $x \notin L \Rightarrow$ every IS in G_x has size $\leq s \cdot 2^{k \log n}$

IS Inapproximability

Let $L \in \mathbf{PCP}_{c,s}[O(\log n), O(1)]$. Configuration is the computation of the Verifier for random bits r and queries q ($\rightarrow 2^q 2^{k \log n}$ configurations). Consider the graph of accepting configurations G_x, where there is an edge between every inconsistent configurations.

- $x \in L \Rightarrow$ there is an IS in G_x of size $\geq c \cdot 2^{k \log n}$
- $x \notin L \Rightarrow$ every IS in G_x has size $\leq s \cdot 2^{k \log n}$

IS Inapproximability

Theorem

There is no 2-approximation algorithm for MAX-IS, unless $\mathbf{P} = \mathbf{NP}$.

Theorem

There is a constant $c > 1$ such than there is no n^c-approximation algorithm for MAX-IS, unless $\mathbf{P} = \mathbf{NP}$.

IS Inapproximability

Theorem

There is no 2-approximation algorithm for MAX-IS, unless $\mathbf{P} = \mathbf{NP}$.

Theorem

There is a constant $c > 1$ such than there is no n^c-approximation algorithm for MAX-IS, unless $\mathbf{P} = \mathbf{NP}$.

Outline

Optimized PCP

Theorem (Håstad '01)

$\mathbf{NP} = \mathbf{PCP}_{1-\epsilon,1/2+\epsilon}[\mathsf{O}(\log n), 3]$ *with a 3-XOR verifier.*

Tight Inapproximability for MAX-3SAT

An XOR is represented by a 4-clause CNF formula.

- $x \in L \Rightarrow \mathbf{Pr}[\text{V accepts}] \geq 1 - \epsilon \Rightarrow (4 - \epsilon)n^k$ satified clauses
- $x \notin L \Rightarrow \mathbf{Pr}[\text{V accepts}] \leq 1/2 + \epsilon \Rightarrow (3.5 - \epsilon)n^k$ satified clauses

Theorem

There is no ρ-approximation for MAX-3SAT with $\rho < 8/7$ unless $\mathbf{P} = \mathbf{NP}$.

More Results

Theorem

There is no ρ-approximation for MAX-CUT with $\rho < 17/16$ unless $\mathbf{P} = \mathbf{NP}$.

Theorem

There is no $n^{1-\epsilon}$-approximation for MAX-IS unless $\mathbf{ZPP} = \mathbf{NP}$.

More results using the Unique Games Conjecture (Khot '02).

Outline

Bibliography

- Inapproximability of Combinatorial Optimization Problems, Luca Trevisan (2004)
- Subhash Khot's Lecture Notes on PCPs and Hardness of approximation (2008)
- Some optimal inapproximability results, Johan Håstad (2001)