BILL AND NATHAN, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

Lower Bounds on Approx for Set Cover

Approximating Set Cover

Set Cover Given n and $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$ find the least number of sets S_i's that **cover** $\{1, \ldots, n\}$.

Approximating Set Cover

Set Cover Given n and $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$ find the least number of sets S_i's that **cover** $\{1, \ldots, n\}$.

1. Chvatal in 1979 showed that there is a poly time approx algorithm for **Set Cover** that will return $(\ln n) \times \mathrm{OPTIMAL}$.

Approximating Set Cover

Set Cover Given n and $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$ find the least number of sets S_i's that **cover** $\{1, \ldots, n\}$.

1. Chvatal in 1979 showed that there is a poly time approx algorithm for **Set Cover** that will return $(\ln n) \times \mathrm{OPTIMAL}$.
2. Dinur and Steurer in 2013 showed that, assuming $\mathrm{P} \neq \mathrm{NP}$, for all ϵ there is no $(1-\epsilon)\ln n \times \mathrm{OPTIMAL}$ approx alg for **Set Cover**

Approximating Set Cover

Set Cover Given n and $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$ find the least number of sets S_i's that **cover** $\{1, \ldots, n\}$.

1. Chvatal in 1979 showed that there is a poly time approx algorithm for **Set Cover** that will return $(\ln n) \times \mathrm{OPTIMAL}$.
2. Dinur and Steurer in 2013 showed that, assuming $\mathrm{P} \neq \mathrm{NP}$, for all ϵ there is no $(1-\epsilon)\ln n \times \mathrm{OPTIMAL}$ approx alg for **Set Cover**

We will **sketch** a proof of a **weaker** lower bound on Set Cover.

2-Prover 1-Round Protocols

Recall PCP

$\boldsymbol{A} \in \mathrm{PCP}(\boldsymbol{q(n)}, \boldsymbol{r(n)}, \boldsymbol{\epsilon(n)})$ if there exists a $q(n)$-query, $r(n)$-random RPOTM-BA $M^{()}$ such that, for all n, for all $x \in \{0,1\}^n$, the following holds.

Recall PCP

$A \in \mathrm{PCP}(q(n), r(n), \epsilon(n))$ if there exists a $q(n)$-query, $r(n)$-random RPOTM-BA $M^{()}$ such that, for all n, for all $x \in \{0,1\}^n$, the following holds.

1. If $x \in A$ then there exists y such that, for all τ with $|\tau| = r(n)$, $M^y(x, \tau)$ accepts. In other words, the probability of acceptance is 1.

Recall PCP

$A \in \mathrm{PCP}(q(n), r(n), \epsilon(n))$ if there exists a $q(n)$-query, $r(n)$-random RPOTM-BA $M^{()}$ such that, for all n, for all $x \in \{0,1\}^n$, the following holds.

1. If $x \in A$ then there exists y such that, for all τ with $|\tau| = r(n)$, $M^y(x, \tau)$ accepts. In other words, the probability of acceptance is 1.
2. If $x \notin A$ then for all y at most $\epsilon(n)$ of the τ's with $|\tau| = r(n)$ make $M^y(x, \tau)$ accept. In other words, the probability of acceptance is $\leq \epsilon(n)$.

Recall PCP

$A \in \mathrm{PCP}(q(n), r(n), \epsilon(n))$ if there exists a $q(n)$-query, $r(n)$-random RPOTM-BA $M^{()}$ such that, for all n, for all $x \in \{0,1\}^n$, the following holds.

1. If $x \in A$ then there exists y such that, for all τ with $|\tau| = r(n)$, $M^y(x,\tau)$ accepts. In other words, the probability of acceptance is 1.
2. If $x \notin A$ then for all y at most $\epsilon(n)$ of the τ's with $|\tau| = r(n)$ make $M^y(x,\tau)$ accept. In other words, the probability of acceptance is $\leq \epsilon(n)$.
3. One of the two cases above must happen.

Aspect of PCP we will Vary

View PCP as a Verifier V interacting with a Prover P.

Aspect of PCP we will Vary

View PCP as a Verifier V interacting with a Prover P.
Note that

Aspect of PCP we will Vary

View PCP as a Verifier V interacting with a Prover P.
Note that
(1) V's queries are adaptive. Can ask one, get the answer, then ask another one.

Aspect of PCP we will Vary

View PCP as a Verifier V interacting with a Prover P.
Note that
(1) V's queries are adaptive. Can ask one, get the answer, then ask another one.
(2) V's queries are bit-queries. $\Sigma = \{0, 1\}$.

Aspect of PCP we will Vary

View PCP as a Verifier V interacting with a Prover P.
Note that
(1) V's queries are adaptive. Can ask one, get the answer, then ask another one.
(2) V's queries are bit-queries. $\Sigma = \{0, 1\}$.
(3) V has 1-sided error.

Aspect of PCP we will Vary

View PCP as a Verifier V interacting with a Prover P.
Note that
(1) V's queries are adaptive. Can ask one, get the answer, then ask another one.
(2) V's queries are bit-queries. $\Sigma = \{0, 1\}$.
(3) V has 1-sided error.
(4) V makes his bit-queries to ONE Prover.

An Actual Example that Will be Relevant

Before defining our new concept formally we will do

An Actual Example that Will be Relevant

Before defining our new concept formally we will do **an example!**

An Actual Example that Will be Relevant

Before defining our new concept formally we will do **an example!**

The example points both backwards and forwards.

An Actual Example that Will be Relevant

Before defining our new concept formally we will do **an example!**

The example points both backwards and forwards.

1. It is similar to the **educational** example I gave of PCP

An Actual Example that Will be Relevant

Before defining our new concept formally we will do **an example!**

The example points both backwards and forwards.

1. It is similar to the **educational** example I gave of PCP
2. We will **use** this protocol later in our lower bound proof for SET COVER.

Prepping for an Example (really!) of a . . .

Recall that we have a gap reduction from 3SAT to MAX3SAT.

Prepping for an Example (really!) of a . . .

Recall that we have a gap reduction from 3SAT to MAX3SAT.
ϕ maps to ϕ' which has m' clauses.

Prepping for an Example (really!) of a . . .

Recall that we have a gap reduction from 3SAT to MAX3SAT.
ϕ maps to ϕ' which has m' clauses.

1. If $\phi \in \mathrm{3SAT}$ then $\mathrm{OPT}(\phi') = m'$.

Prepping for an Example (really!) of a ...

Recall that we have a gap reduction from 3SAT to MAX3SAT.
ϕ maps to ϕ' which has m' clauses.

1. If $\phi \in \mathrm{3SAT}$ then $\mathrm{OPT}(\phi') = m'$.
2. If $\phi \notin \mathrm{3SAT}$ then $\mathrm{OPT}(\phi') \leq (1 - \delta)m'$.

Prepping for an Example (really!) of a ...

Recall that we have a gap reduction from 3SAT to MAX3SAT.
ϕ maps to ϕ' which has m' clauses.

1. If $\phi \in \mathrm{3SAT}$ then $\mathrm{OPT}(\phi') = m'$.
2. If $\phi \notin \mathrm{3SAT}$ then $\mathrm{OPT}(\phi') \le (1-\delta)m'$.

It is of interest to look at formulas ψ which we are promised are either **satisfiable** or **far from satisfiable**.

More Prepping for an Example (really!) of a ...

There is a gap reduction from MAX3SAT to MAX3SAT-5. (We will see this in a later talk.)

More Prepping for an Example (really!) of a ...

There is a gap reduction from $\mathrm{MAX3SAT}$ to $\mathrm{MAX3SAT\text{-}5}$. (We will see this in a later talk.)
ϕ' maps to ψ which has m clauses.

More Prepping for an Example (really!) of a ...

There is a gap reduction from $\mathrm{MAX3SAT}$ to $\mathrm{MAX3SAT\text{-}5}$. (We will see this in a later talk.)
ϕ' maps to ψ which has m clauses.

1. If $\phi' \in \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) = m$.

More Prepping for an Example (really!) of a ...

There is a gap reduction from $\mathrm{MAX3SAT}$ to $\mathrm{MAX3SAT\text{-}5}$. (We will see this in a later talk.)

ϕ' maps to ψ which has m clauses.

1. If $\phi' \in \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) = m$.
2. If $\phi' \notin \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) \le (1 - \delta)m$.

More Prepping for an Example (really!) of a ...

There is a gap reduction from $\mathrm{MAX3SAT}$ to $\mathrm{MAX3SAT\text{-}5}$. (We will see this in a later talk.)
ϕ' maps to ψ which has m clauses.

1. If $\phi' \in \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) = m$.
2. If $\phi' \notin \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) \leq (1-\delta)m$.
3. Every variable in ψ appears exactly 5 times. Important for us: $m = \Theta(n)$.

More Prepping for an Example (really!) of a ...

There is a gap reduction from $\mathrm{MAX3SAT}$ to $\mathrm{MAX3SAT\text{-}5}$. (We will see this in a later talk.)
ϕ' maps to ψ which has m clauses.

1. If $\phi' \in \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) = m$.
2. If $\phi' \notin \mathrm{3SAT}$ then $\mathrm{OPT}(\psi) \le (1-\delta)m$.
3. Every variable in ψ appears exactly 5 times. Important for us: $m = \Theta(n)$.

It is of interest to look at formulas ψ which we are promised are either **satisfiable** or **far from satisfiable** and where $m = \Theta(n)$.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.
2. V picks a random clause $C = (L_1 \vee L_2 \vee L_3)$ and a random L_i from it. This takes $O(\log m) = O(\log n)$ random bits.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.
2. V picks a random clause $C = (L_1 \vee L_2 \vee L_3)$ and a random L_i from it. This takes $O(\log m) = O(\log n)$ random bits.
3. V asks P1 the truth-assignment for L_i (1 bit) and P2 the truth-assignment for (L_1, L_2, L_3) (3 bits).

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.
2. V picks a random clause $C = (L_1 \vee L_2 \vee L_3)$ and a random L_i from it. This takes $O(\log m) = O(\log n)$ random bits.
3. V asks P1 the truth-assignment for L_i (1 bit) and P2 the truth-assignment for (L_1, L_2, L_3) (3 bits).
4. If answers are consistent and make the clause T, then V accepts, else V rejects.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.
2. V picks a random clause $C = (L_1 \vee L_2 \vee L_3)$ and a random L_i from it. This takes $O(\log m) = O(\log n)$ random bits.
3. V asks P1 the truth-assignment for L_i (1 bit) and P2 the truth-assignment for (L_1, L_2, L_3) (3 bits).
4. If answers are consistent and make the clause T, then V accepts, else V rejects.

Note
(1) Query to P_2 is considered ONE query where alphabet is $\{0,1\}^3$.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \leq (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.
2. V picks a random clause $C = (L_1 \vee L_2 \vee L_3)$ and a random L_i from it. This takes $O(\log m) = O(\log n)$ random bits.
3. V asks P1 the truth-assignment for L_i (1 bit) and P2 the truth-assignment for (L_1, L_2, L_3) (3 bits).
4. If answers are consistent and make the clause T, then V accepts, else V rejects.

Note

(1) Query to P_2 is considered ONE query where alphabet is $\{0, 1\}^3$.
(2) Two provers P_1, P_2 cannot communicate.

Example of a BLAH for Promise-MAX3SAT

V and **two** P_1, P_2 are looking at ψ which has m clauses.
V is promised that either $\mathrm{OPT}(\psi) = m$ or $\mathrm{OPT}(\psi) \le (1-\delta)m$.

1. Input 3CNF ψ: either 3SAT or far from 3SAT, $m = \Theta(n)$.
2. V picks a random clause $C = (L_1 \vee L_2 \vee L_3)$ and a random L_i from it. This takes $O(\log m) = O(\log n)$ random bits.
3. V asks P1 the truth-assignment for L_i (1 bit) and P2 the truth-assignment for (L_1, L_2, L_3) (3 bits).
4. If answers are consistent and make the clause T, then V accepts, else V rejects.

Note
(1) Query to P_2 is considered ONE query where alphabet is $\{0,1\}^3$.
(2) Two provers P_1, P_2 cannot communicate.
(3) When V gets the answers he will then decide if he thinks $\psi \in \mathrm{3SAT}$.