What is this Unique Games Conjecture that everyone keeps talking about?

Per Austrin
austrin@kth.se

Theory Group
School of Computer Science and Communication
KTH – Royal Institute of Technology, Stockholm

2007-02-05

Agenda

1. The conjecture

2. Implications
   - Assorted graph problems
   - Constraint satisfaction

3. Progress?

Agenda

1. The conjecture
2. Implications
   - Assorted graph problems
   - Constraint satisfaction
3. Progress?

The Label Cover Problem

- In the Label Cover problem we are given
  - A graph $G = (V, E)$
  - A set of “labels” L
  - For each edge $e = (v, w) \in E$, a set $R_e \subseteq L \times L$ consisting of a set of “permissible” values for the pair (v, w)

The Label Cover Problem

- In the Label Cover problem we are given
  - A graph $G = (V, E)$
  - A set of "labels" L
  - For each edge $e = (v, w) \in E$, a set $R_e \subseteq L \times L$ consisting of a set of "permissible" values for the pair (v, w)
- Goal: assign a label $l_v \in L$ to each vertex $v \in V$ such that as many edges as possible are satisfied

The Label Cover Problem

- In the Label Cover problem we are given
  - A graph $G = (V, E)$
  - A set of “labels” L
  - For each edge $e = (v, w) \in E$, a set $R_e \subseteq L \times L$ consisting of a set of “permissible” values for the pair (v, w)
- Goal: assign a label $l_v \in L$ to each vertex $v \in V$ such that as many edges as possible are satisfied
- An edge $e = (v, w)$ is satisfied if $(l_v, l_w) \in R_e$.

The Label Cover Problem

- In the Label Cover problem we are given
  - A graph $G = (V, E)$
  - A set of "labels" L
  - For each edge $e = (v, w) \in E$, a set $R_e \subseteq L \times L$ consisting of a set of "permissible" values for the pair (v, w)
- Goal: assign a label $l_v \in L$ to each vertex $v \in V$ such that as many edges as possible are satisfied
- An edge $e = (v, w)$ is satisfied if $(l_v, l_w) \in R_e$.
- We denote by $\mathrm{Val}(X) \in [0, 1]$ the maximum possible fraction of satisfied edges.

Example

We can view 3-coloring as a Label Cover problem.

Example

We can view 3-coloring as a Label Cover problem.

- Same graph.
- $L = \{1, 2, 3\}$
- $R_e = \{ (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2) \}$ for all edges.

Example

We can view 3-coloring as a Label Cover problem.

- Same graph.
- $L = \{1, 2, 3\}$
- $R_e = \{\, (1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2) \,\}$ for all edges.
- $\mathrm{Val}(X) = 1 \Leftrightarrow G$ is 3-colorable.

Example

We can view 3-coloring as a Label Cover problem.

- Same graph.
- $L = \{1, 2, 3\}$
- $R_e = \{\,(1,2),(1,3),(2,1),(2,3),(3,1),(3,2)\,\}$ for all edges.
- $\mathrm{Val}(X) = 1 \Leftrightarrow G$ is 3-colorable.

Thus it is NP-hard to determine if we can satisfy all edges.

Another example

- Consider a system of linear equations mod n, where each equation contains exactly 2 variables.

Another example

- Consider a system of linear equations mod n, where each equation contains exactly 2 variables.
- Goal: satisfy as many equations as possible.

Another example

- Consider a system of linear equations mod n, where each equation contains exactly 2 variables.
- Goal: satisfy as many equations as possible.
- This is an instance of Label Cover with label set $L = \mathbb{Z}_n$.

Yet Another example

We can view 3-satisfiability as a Label Cover problem.

Yet Another example

We can view 3-satisfiability as a Label Cover problem.

- $V = \text{Clauses} \cup \text{Variables}$

Yet Another example

We can view 3-satisfiability as a Label Cover problem.

- V = Clauses $\cup$ Variables
- Edge between clause ϕ and variable x if x occurs in ϕ.

Yet Another example

We can view 3-satisfiability as a Label Cover problem.

- V = Clauses $\cup$ Variables
- Edge between clause ϕ and variable x if x occurs in ϕ.
- $L = \{000, 001, 010, 011, 100, 101, 110, 111, 0, 1\}$

Yet Another example

We can view 3-satisfiability as a Label Cover problem.

- $V =$ Clauses $\cup$ Variables
- Edge between clause ϕ and variable x if x occurs in ϕ.
- $L = \{000, 001, 010, 011, 100, 101, 110, 111, 0, 1\}$
- Edge between ϕ and x satisfied if
  - label for ϕ is one of the seven satisfying assignments for the three variables in ϕ.
  - labels for x and ϕ give the same value to x.

Yet Another example

We can view 3-satisfiability as a Label Cover problem.

- $V =$ Clauses $\cup$ Variables
- Edge between clause ϕ and variable x if x occurs in ϕ.
- $L = \{000, 001, 010, 011, 100, 101, 110, 111, 0, 1\}$
- Edge between ϕ and x satisfied if
  - label for ϕ is one of the seven satisfying assignments for the three variables in ϕ.
  - labels for x and ϕ give the same value to x.

Does this seem familiar?

Two-prover games

- Consider a two-prover game:
  - two provers try to convince us (the verifier) that some 3-CNF formula is satisfiable
  - we send a question to each of the provers, which they have to answer independently

Two-prover games

- Consider a two-prover game:
  - two provers try to convince us (the verifier) that some 3-CNF formula is satisfiable
  - we send a question to each of the provers, which they have to answer independently
- We can view this as a Label Cover problem

Two-prover games

- Consider a two-prover game:
  - two provers try to convince us (the verifier) that some 3-CNF formula is satisfiable
  - we send a question to each of the provers, which they have to answer independently
- We can view this as a Label Cover problem
  - $V =$ the set of possible questions
  - $L =$ the set of possible answers
  - An edge (v, w) is satisfied by a pair of answers if that pair of answers make the verifier accept

Two-prover games

- Consider a two-prover game:
  - two provers try to convince us (the verifier) that some 3-CNF formula is satisfiable
  - we send a question to each of the provers, which they have to answer independently
- We can view this as a Label Cover problem
  - $V =$ the set of possible questions
  - $L =$ the set of possible answers
  - An edge (v, w) is satisfied by a pair of answers if that pair of answers make the verifier accept
- Finding a good strategy for the provers is equivalent to solving the label cover problem

Approximating Label Cover

Theorem

For every $\eta > 0$ it is NP-hard to distinguish Label Cover instances with $\mathrm{Val}(X) = 1$ from those with $\mathrm{Val}(X) \leq \eta$

Approximating Label Cover

Theorem

For every $\eta > 0$ there is an L such that it is NP-hard to distinguish Label Cover instances with $\mathrm{Val}(X) = 1$ from those with $\mathrm{Val}(X) \leq \eta$ for X with label set L

Approximating Label Cover

Theorem

For every $\eta > 0$ there is an L such that it is NP-hard to distinguish Label Cover instances with $\text{Val}(X) = 1$ from those with $\text{Val}(X) \leq \eta$ for X with label set L

(Follows from PCP theorem and Parallel Repetition theorem)

Unique Label Cover

- Unique Label Cover is the special case of Label Cover when the relation for each edge is a permutation

Unique Label Cover

- Unique Label Cover is the special case of Label Cover when the relation for each edge is a permutation
  - In other words, for each edge $e = (v, w)$ and choice of label for v there is exactly one choice of label for w that satisfies the edge (and vice versa)

Unique Label Cover

- Unique Label Cover is the special case of Label Cover when the relation for each edge is a permutation
  - In other words, for each edge $e = (v, w)$ and choice of label for v there is exactly one choice of label for w that satisfies the edge (and vice versa)
  - In the 2-prover game, given the answer from one of the provers, there is exactly one answer from the other prover that will make the verifier accept

Unique Label Cover

- Unique Label Cover is the special case of Label Cover when the relation for each edge is a permutation
  - In other words, for each edge $e = (v, w)$ and choice of label for v there is exactly one choice of label for w that satisfies the edge (and vice versa)
  - In the 2-prover game, given the answer from one of the provers, there is exactly one answer from the other prover that will make the verifier accept
- Our example with linear equations is a Unique Label Cover problem if n is prime

Unique Label Cover

- Unique Label Cover is the special case of Label Cover when the relation for each edge is a permutation
  - In other words, for each edge $e = (v, w)$ and choice of label for v there is exactly one choice of label for w that satisfies the edge (and vice versa)
  - In the 2-prover game, given the answer from one of the provers, there is exactly one answer from the other prover that will make the verifier accept
- Our example with linear equations is a Unique Label Cover problem if n is prime
- How hard is Unique Label Cover?

The Unique Games Conjecture

Conjecture (Khot, 2002)

For any $\eta > 0$ *there is an L such that it is NP-hard to distinguish Unique Label Cover instances with* $\mathrm{Val}(X) \geq 1 - \eta$ *from those with* $\mathrm{Val}(X) \leq \eta$ *for* X *with label set L.*

Agenda

But first some terminology

- Approximation ratio of an algorithm is expected value of solution found divided by optimum value

But first some terminology

- Approximation ratio of an algorithm is expected value of solution found divided by optimum value
  - ≤ 1 for maximization problems
  - ≥ 1 for minimization problems

But first some terminology

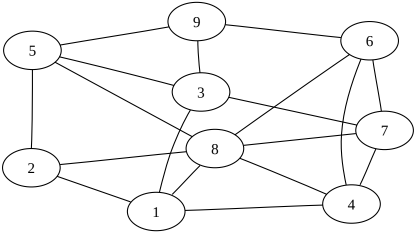
- Approximation ratio of an algorithm is expected value of solution found divided by optimum value
  - ≤ 1 for maximization problems
  - ≥ 1 for minimization problems
- A problem $\mathcal{P}$ is "UG-hard" if Unique Label Cover can be "efficiently" reduced to $\mathcal{P}$

But first some terminology

- Approximation ratio of an algorithm is expected value of solution found divided by optimum value
  - ≤ 1 for maximization problems
  - ≥ 1 for minimization problems
- A problem $\mathcal{P}$ is "UG-hard" if Unique Label Cover can be "efficiently" reduced to $\mathcal{P}$
- UG-hard $\Rightarrow$ NP-hard assuming the UGC

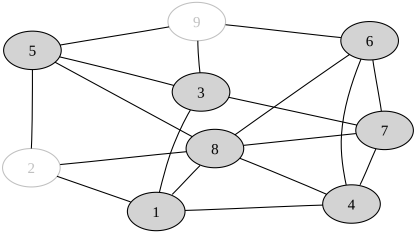
Min Vertex Cover

- Given a graph $G = (V, E)$, a set $C \subseteq V$ is a vertex cover if for each edge $\{v, w\}$, at least one of v and w is in C



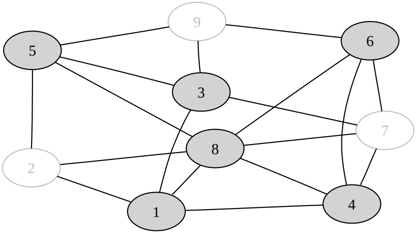
Min Vertex Cover

- Given a graph $G = (V, E)$, a set $C \subseteq V$ is a vertex cover if for each edge $\{v, w\}$, at least one of v and w is in C



Min Vertex Cover

- Given a graph $G = (V, E)$, a set $C \subseteq V$ is a vertex cover if for each edge $\{v, w\}$, at least one of v and w is in C
- Goal: find a vertex cover of minimum cardinality



Min Vertex Cover

- Easy to approximate within a factor 2

Min Vertex Cover

- Easy to approximate within a factor 2
  - While there are edges which are not covered, include both vertices of one such edge in the cover

Min Vertex Cover

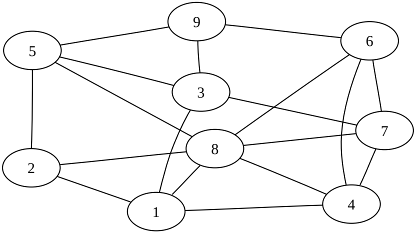
- Easy to approximate within a factor 2
    - While there are edges which are not covered, include both vertices of one such edge in the cover
- Dinur and Safra, 2002: NP-hard to approximate within 1.36

Min Vertex Cover

- Easy to approximate within a factor 2
  - While there are edges which are not covered, include both vertices of one such edge in the cover
- Dinur and Safra, 2002: NP-hard to approximate within 1.36
- Khot and Regev, 2003: UG-hard to approximate within $2 - \epsilon$ for any $\epsilon > 0$

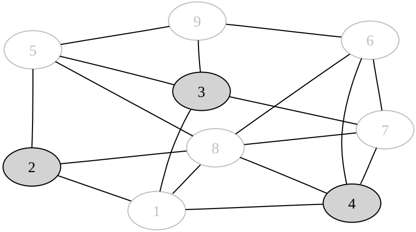
Max Independent Set

- Consider the Maximum Independent Set problem restricted to graphs where every vertex has degree $\leq \Delta$



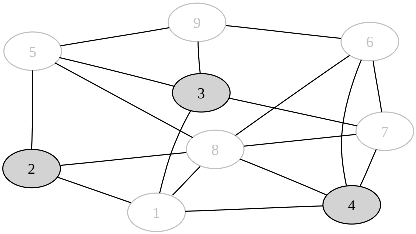
Max Independent Set

- Consider the Maximum Independent Set problem restricted to graphs where every vertex has degree $\leq \Delta$



Max Independent Set

- Consider the Maximum Independent Set problem restricted to graphs where every vertex has degree $\leq \Delta$
  - Think of Δ as big, but constant
  - NP-hard for any constant $\Delta \geq 3$



Max Independent Set

- Easy to approximate within $1/(\Delta + 1)$

Max Independent Set

- Easy to approximate within $1/(\Delta+1)$
- Vishwanathan, unpublished: Can be approximated within $\frac{c\log\Delta}{\log\log\Delta}\cdot 1/\Delta$ for some $c>0$

Max Independent Set

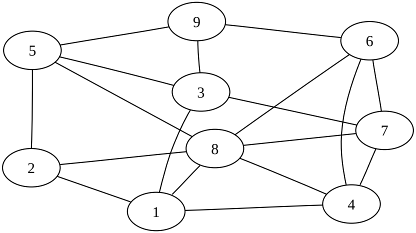
- Easy to approximate within $1/(\Delta + 1)$
- Vishwanathan, unpublished: Can be approximated within $\frac{c \log \Delta}{\log \log \Delta} \cdot 1/\Delta$ for some $c > 0$
- Trevisan, 2001: NP-hard to approximate within $2^{c\sqrt{\log \Delta}}/\Delta$ for some $c > 0$

Max Independent Set

- Easy to approximate within $1/(\Delta+1)$
- Vishwanathan, unpublished: Can be approximated within $\frac{c \log \Delta}{\log \log \Delta} \cdot 1/\Delta$ for some $c > 0$
- Trevisan, 2001: NP-hard to approximate within $2^{c\sqrt{\log \Delta}}/\Delta$ for some $c > 0$
- Samorodnitsky and Trevisan, 2005: UG-hard to approximate within $(\log \Delta)^c/\Delta$ for some constant c

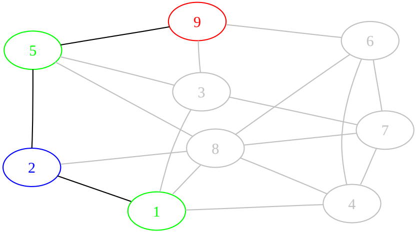
Almost 3-coloring

- Given a graph $G = (V, E)$, we want to remove some vertices and get a 3-colorable graph



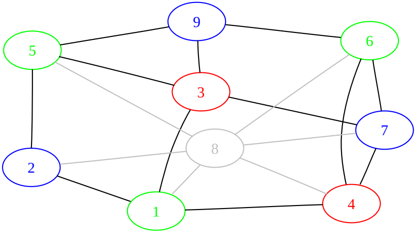
Almost 3-coloring

- Given a graph $G = (V, E)$, we want to remove some vertices and get a 3-colorable graph



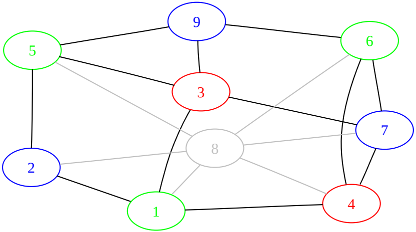
Almost 3-coloring

- Given a graph $G = (V, E)$, we want to remove some vertices and get a 3-colorable graph
- Goal: remove as few vertices as possible
  - Call this number $R(G)$



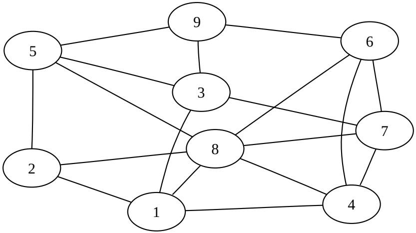
Almost 3-coloring

- Given a graph $G = (V, E)$, we want to remove some vertices and get a 3-colorable graph
- Goal: remove as few vertices as possible
  - Call this number $R(G)$
- Dinur, Mossel, and Regev, 2005: UG-hard to distinguish between $R(G) \leq \epsilon|V|$ and $R(G) \geq (1-\epsilon)|V|$ for any $\epsilon > 0$



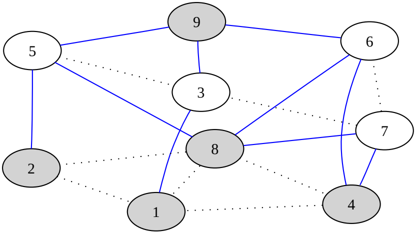
Max Cut

- Given a graph $G = (V, E)$ and a set $S \subseteq V$, let $|E(S, \overline{S})|$ denote the number of edges cut by S



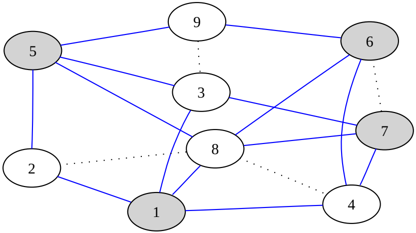
Max Cut

- Given a graph $G = (V, E)$ and a set $S \subseteq V$, let $|E(S, \overline{S})|$ denote the number of edges cut by S



Max Cut

- Given a graph $G = (V, E)$ and a set $S \subseteq V$, let $|E(S, \overline{S})|$ denote the number of edges cut by S
- Goal: maximize number of edges cut



Max Cut

- Easy to approximate within a factor $1/2$

Max Cut

- Easy to approximate within a factor $1/2$
- Goemans and Williamson, 1995: Max Cut can be approximated to within a factor $\alpha_{GW} \approx 0.8785$

Max Cut

- Easy to approximate within a factor $1/2$
- Goemans and Williamson, 1995: Max Cut can be approximated to within a factor $\alpha_{GW} \approx 0.8785$
- Håstad, 2001: NP-hard to approximate within $16/17 + \epsilon \approx 0.9418$

Max Cut

- Easy to approximate within a factor $1/2$
- Goemans and Williamson, 1995: Max Cut can be approximated to within a factor $\alpha_{GW} \approx 0.8785$
- Håstad, 2001: NP-hard to approximate within $16/17 + \epsilon \approx 0.9418$
- Khot, Kindler, Mossel and O'Donnell, 2004: UG-hard to approximate within $\alpha_{GW} + \epsilon$ for any ϵ

Max 2-Sat

- Given a 2-CNF formula, find an assignment which satisfies as many clauses as possible

$$
\begin{array}{llllll}
(\overline{x_1} \vee x_2) & \wedge & (x_1 \vee \overline{x_3}) & \wedge & (x_1 \vee x_4) & \wedge \\
(\overline{x_1} \vee x_8) & \wedge & (\overline{x_2} \vee x_5) & \wedge & (\overline{x_2} \vee \overline{x_8}) & \wedge \\
(x_3 \vee x_5) & \wedge & (x_3 \vee x_7) & \wedge & (\overline{x_3} \vee \overline{x_9}) & \wedge \\
(\overline{x_4} \vee x_6) & \wedge & (x_4 \vee \overline{x_7}) & \wedge & (\overline{x_4} \vee \overline{x_8}) & \wedge \\
(\overline{x_5} \vee x_8) & \wedge & (x_5 \vee x_9) & \wedge & (\overline{x_6} \vee \overline{x_7}) & \wedge \\
(x_6 \vee \overline{x_8}) & \wedge & (\overline{x_6} \vee x_9) & \wedge & (x_7 \vee \overline{x_8}) &
\end{array}
$$

Max 2-Sat

- Given a 2-CNF formula, find an assignment which satisfies as many clauses as possible

$$
\begin{array}{ccccccc}
(\overline{x_1} \vee x_2) & \wedge & (x_1 \vee \overline{x_3}) & \wedge & (x_1 \vee x_4) & \wedge \\
(\overline{x_1} \vee x_8) & \wedge & (\overline{x_2} \vee x_5) & \wedge & (\overline{x_2} \vee \overline{x_8}) & \wedge \\
(x_3 \vee x_5) & \wedge & (x_3 \vee x_7) & \wedge & (\overline{x_3} \vee \overline{x_9}) & \wedge \\
(\overline{x_4} \vee x_6) & \wedge & (x_4 \vee \overline{x_7}) & \wedge & (\overline{x_4} \vee \overline{x_8}) & \wedge \\
(\overline{x_5} \vee x_8) & \wedge & (x_5 \vee x_9) & \wedge & (\overline{x_6} \vee \overline{x_7}) & \wedge \\
(x_6 \vee \overline{x_8}) & \wedge & (\overline{x_6} \vee x_9) & \wedge & (x_7 \vee \overline{x_8}) &
\end{array}
$$

$$
\begin{aligned}
x_1 &= \textsc{True} \\
x_2 &= \textsc{True} \\
x_3 &= \textsc{True} \\
x_4 &= \textsc{False} \\
x_5 &= \textsc{True} \\
x_6 &= \textsc{False} \\
x_7 &= \textsc{False} \\
x_8 &= \textsc{False} \\
x_9 &= \textsc{False}
\end{aligned}
$$

Max 2-Sat

- Easy to approximate within a factor $3/4$

Max 2-Sat

- Easy to approximate within a factor $3/4$
- Lewin, Livnat and Zwick, 2002: Max 2-Sat can *apparently* be approximated to within a factor $\alpha_{LLZ} \approx 0.9401$

Max 2-Sat

- Easy to approximate within a factor $3/4$
- Lewin, Livnat and Zwick, 2002: Max 2-Sat can *apparently* be approximated to within a factor $\alpha_{LLZ} \approx 0.9401$
- Håstad, 2001: NP-hard to approximate within $21/22 + \epsilon \approx 0.9546$

Max 2-Sat

- Easy to approximate within a factor $3/4$
- Lewin, Livnat and Zwick, 2002: Max 2-Sat can *apparently* be approximated to within a factor $\alpha_{LLZ} \approx 0.9401$
- Håstad, 2001: NP-hard to approximate within $21/22 + \epsilon \approx 0.9546$
- Austrin, 2006: UG-hard to approximate within $\alpha_{LLZ} + \epsilon$ for any ϵ

Max k-CSP

- In the Max k-CSP problem we are given a set of constraints, each of which acts on at most k variables

$$
\begin{array}{lcll}
((x_1 \vee \overline{x_2}) \wedge x_4) & \wedge & (x_3 \oplus x_4 \oplus x_5) & \wedge \\
(x_1 \vee x_4) & \wedge & (x_2 \wedge \overline{x}_4 \wedge x_1) & \wedge \\
(x_2 \vee (x_3 \oplus x_5)) & \wedge & (\overline{x_1} \wedge \overline{x_3}) &
\end{array}
$$

Max k-CSP

- In the Max k-CSP problem we are given a set of constraints, each of which acts on at most k variables
- Goal: satisfy as many constraints as possible

$$\begin{array}{lll} ((x_1 \vee \overline{x_2}) \wedge x_4) & \wedge & (x_3 \oplus x_4 \oplus x_5) \quad \wedge \\ (x_1 \vee x_4) & \wedge & (x_2 \wedge \overline{x}_4 \wedge x_1) \quad \wedge \\ (x_2 \vee (x_3 \oplus x_5)) & \wedge & (\overline{x_1} \wedge \overline{x_3}) \end{array}$$

$$\begin{array}{lll} x_1 & = & \text{FALSE} \\ x_2 & = & \text{FALSE} \\ x_3 & = & \text{FALSE} \\ x_4 & = & \text{TRUE} \\ x_5 & = & \text{TRUE} \end{array}$$

Max k-CSP

- Easy to approximate within $1/2^k$

Max k-CSP

- Easy to approximate within $1/2^k$
- Charikar, Makarychev and Makarychev, 2006: Max k-CSP can be approximated within $0.44k/2^k$

Max k-CSP

- Easy to approximate within $1/2^k$
- Charikar, Makarychev and Makarychev, 2006: Max k-CSP can be approximated within $0.44k/2^k$
- Engebretsen and Holmerin, 2005: NP-hard to approximate within $2^{\sqrt{2k}}/2^k$

Max k-CSP

- Easy to approximate within $1/2^k$
- Charikar, Makarychev and Makarychev, 2006: Max k-CSP can be approximated within $0.44k/2^k$
- Engebretsen and Holmerin, 2005: NP-hard to approximate within $2^{\sqrt{2k}}/2^k$
- Samorodnitsky and Trevisan, 2005: UG-hard to approximate within $2k/2^k$

Max k-CSP

- Easy to approximate within $1/2^k$
- Charikar, Makarychev and Makarychev, 2006: Max k-CSP can be approximated within $0.44k/2^k$
- Engebretsen and Holmerin, 2005: NP-hard to approximate within $2^{\sqrt{2k}}/2^k$
- Samorodnitsky and Trevisan, 2005: UG-hard to approximate within $2k/2^k$
  - For $k = 2^d - 1$, hard to approximate within $(k+1)/2^k$

Random Max k-CSP

- Pick a random predicate P on k variables by, for each input $x \in \{0,1\}^k$ letting $P(x) = 1$ with probability q

Random Max k-CSP

- Pick a random predicate P on k variables by, for each input $x \in \{0,1\}^k$ letting $P(x) = 1$ with probability q
- Håstad, unpublished: If $q \geq (1/k)^{1/2+\epsilon}$, the probability that it is UG-hard to approximate P better than a random assignment tends to 1 as $k \to \infty$

Random Max k-CSP

- Pick a random predicate P on k variables by, for each input $x \in \{0, 1\}^k$ letting $P(x) = 1$ with probability q
- Håstad, unpublished: If $q \geq (1/k)^{1/2+\epsilon}$, the probability that it is UG-hard to approximate P better than a random assignment tends to 1 as $k \to \infty$
  - For $k = 2^d - 1$, we can take $q \geq (1/k)^{1+\epsilon}$ instead

Max 2-CSP

Max 2-CSP

- Lewin, Livnat and Zwick, 2002: Max 2-CSP can *apparently* be approximated within 0.8740

Max 2-CSP

- Lewin, Livnat and Zwick, 2002: Max 2-CSP can *apparently* be approximated within 0.8740
- Håstad, 2001: NP-hard to approximate within $11/12 + \epsilon \approx 0.9167$

Max 2-CSP

- Lewin, Livnat and Zwick, 2002: Max 2-CSP can *apparently* be approximated within 0.8740
- Håstad, 2001: NP-hard to approximate within $11/12 + \epsilon \approx 0.9167$
- Austrin, unpublished: UG-hard to approximate within 0.8744

2-query proof verification

- Consider extremely lazy verification of proofs, where we only read two bits.

2-query proof verification

- Consider extremely lazy verification of proofs, where we only read two bits.
- Our verification protocol has completeness c and soundness s if

2-query proof verification

- Consider extremely lazy verification of proofs, where we only read two bits.
- Our verification protocol has completeness c and soundness s if
    - for correct statements there is a proof which we accept with probability $\geq c$

2-query proof verification

- Consider extremely lazy verification of proofs, where we only read two bits.
- Our verification protocol has completeness c and soundness s if
  - for correct statements there is a proof which we accept with probability $\geq c$
  - for incorrect statements every proof is accepted with probability $\leq s$

2-query proof verification

- Consider extremely lazy verification of proofs, where we only read two bits.
- Our verification protocol has completeness c and soundness s if
  - for correct statements there is a proof which we accept with probability $\geq c$
  - for incorrect statements every proof is accepted with probability $\leq s$
- How large separation can we get between s and c?

2-query proof verification

- Assuming the UGC (and $P \neq NP$)

2-query proof verification

- Assuming the UGC (and $P \neq NP$)
  - can take $c = 0.4411$ and $s = 0.3858$

2-query proof verification

- Assuming the UGC (and $P \neq NP$)
  - can take $c = 0.4411$ and $s = 0.3858$
  - can take $c = 1 - \epsilon$ and $s = 1 - \Theta(\sqrt{\epsilon})$

2-query proof verification

- Assuming the UGC (and $P \neq NP$)
  - can take $c = 0.4411$ and $s = 0.3858$
  - can take $c = 1 - \epsilon$ and $s = 1 - \Theta(\sqrt{\epsilon})$
- Assuming $P \neq NP$

2-query proof verification

- Assuming the UGC (and $P \neq NP$)
  - can take $c = 0.4411$ and $s = 0.3858$
  - can take $c = 1 - \epsilon$ and $s = 1 - \Theta(\sqrt{\epsilon})$
- Assuming $P \neq NP$
  - can not have $c = 0.4411$ and $s = 0.3855$

2-query proof verification

- Assuming the UGC (and $P \neq NP$)
  - can take $c = 0.4411$ and $s = 0.3858$
  - can take $c = 1 - \epsilon$ and $s = 1 - \Theta(\sqrt{\epsilon})$
- Assuming $P \neq NP$
  - can not have $c = 0.4411$ and $s = 0.3855$
  - can not have $c = 1 - \epsilon$ and $s = 1 - \omega(\sqrt{\epsilon})$

Agenda

1. The conjecture
2. Implications
   - Assorted graph problems
   - Constraint satisfaction
3. Progress?

How large does L have to be?

- Easy to find an assignment which satisfies an $1/L$ fraction of all edges, so must have $L \geq 1/\mu$

How large does L have to be?

- Easy to find an assignment which satisfies an $1/L$ fraction of all edges, so must have $L \geq 1/\mu$
- Khot, Kindler, Mossel, and O'Donnell, 2004: If the UGC is true, we can choose $|L| \leq (1/\eta)^{2/\eta-1}$

How large does L have to be?

- Easy to find an assignment which satisfies an $1/L$ fraction of all edges, so must have $L \geq 1/\mu$
- Khot, Kindler, Mossel, and O'Donnell, 2004: If the UGC is true, we can choose $|L| \leq (1/\eta)^{2/\eta-1}$
  - For $\eta = 0.01$, this is 10^{398}

How large does L have to be?

- Easy to find an assignment which satisfies an $1/L$ fraction of all edges, so must have $L \geq 1/\mu$
- Khot, Kindler, Mossel, and O'Donnell, 2004: If the UGC is true, we can choose $|L| \leq (1/\eta)^{2/\eta - 1}$
  - For $\eta = 0.01$, this is 10^{398}
  - Furthermore, we can use linear equations mod p

How large does L have to be?

- Easy to find an assignment which satisfies an $1/L$ fraction of all edges, so must have $L \geq 1/\mu$
- Khot, Kindler, Mossel, and O'Donnell, 2004: If the UGC is true, we can choose $|L| \leq (1/\eta)^{2/\eta-1}$
  - For $\eta = 0.01$, this is 10^{398}
  - Furthermore, we can use linear equations mod p
- Charikar, Makarychev and Makarychev, 2006: The UGC is *not* true for $|L| \leq (\Omega(1/\eta))^{2/\eta}$.

Inapproximability of Unique Label Cover

Theorem (Feige and Reichman)

For any $\eta > 0$ *there is a* $0 < \gamma < 1$ *and L and such that it is NP-hard to distinguish between Unique Label Cover instances with* $\mathrm{Val}(X) \geq \gamma$ *and* $\mathrm{Val}(X) \leq \eta \cdot \gamma$ *for* X *with label set L.*

Proof (1/3)

Recall

Theorem

For every $\eta > 0$ *there is an L such that it is NP-hard to distinguish Label Cover instances with* $\mathrm{Val}(X) = 1$ *from those with* $\mathrm{Val}(X) \leq \eta$ *for* X *with label set L.*

Proof (1/3)

Recall

Theorem

For every $\eta > 0$ there is an L such that it is NP-hard to distinguish Label Cover instances with $\mathrm{Val}(X) = 1$ from those with $\mathrm{Val}(X) \leq \eta$ for X with label set L.

We'll reduce this problem to linear equations mod p.

Proof (1/3)

Recall

Theorem

For every $\eta > 0$ there is an L such that it is NP-hard to distinguish Label Cover instances with $\mathrm{Val}(X) = 1$ from those with $\mathrm{Val}(X) \leq \eta$ for X with label set L.

We'll reduce this problem to linear equations mod p.

- Pick L as in the theorem, and let $X = (V, E, L, \{R_e\})$ be a Label Cover instance with either $\mathrm{Val}(X) = 1$ or $\mathrm{Val}(X) \leq \eta$.

Proof (1/3)

Recall

Theorem

For every $\eta > 0$ *there is an L such that it is NP-hard to distinguish Label Cover instances with* $\mathrm{Val}(X) = 1$ *from those with* $\mathrm{Val}(X) \leq \eta$ *for X with label set L.*

We'll reduce this problem to linear equations mod p.

- Pick L as in the theorem, and let $X = (V, E, L, \{R_e\})$ be a Label Cover instance with either $\mathrm{Val}(X) = 1$ or $\mathrm{Val}(X) \leq \eta$.
- Take a prime $p \geq 1 + |L|^2/\eta$, and identify the elements of L with elements of $\mathbb{Z}_p$ in some arbitrary way.

Proof (2/3)

For each edge $e = (u, v) \in E$

Proof (2/3)

For each edge $e = (u, v) \in E$

- For each $(a, b) \in R_e$ and $i \in \{1, \ldots, p-1\}$
  - Add the equation $x_u - a = i \cdot (x_v - b)$

Proof (2/3)

For each edge $e = (u, v) \in E$

- For each $(a, b) \in R_e$ and $i \in \{1, \ldots, p-1\}$
  - Add the equation $x_u - a = i \cdot (x_v - b)$
- Note that given an assignment to the x variables

Proof (2/3)

For each edge $e = (u, v) \in E$

- For each $(a, b) \in R_e$ and $i \in \{1, \ldots, p-1\}$
  - Add the equation $x_u - a = i \cdot (x_v - b)$
- Note that given an assignment to the x variables
  - If $(x_u, x_v) \notin R_e$ at most $|R_e| \leq |L|^2 \leq \eta(p-1)$ of the equations are satisfied

Proof (2/3)

For each edge $e = (u, v) \in E$

- For each $(a, b) \in R_e$ and $i \in \{1, \ldots, p-1\}$
  - Add the equation $x_u - a = i \cdot (x_v - b)$
- Note that given an assignment to the x variables
  - If $(x_u, x_v) \notin R_e$ at most $|R_e| \leq |L|^2 \leq \eta(p-1)$ of the equations are satisfied
  - If $(x_u, x_v) \in R_e$

Proof (2/3)

For each edge $e = (u, v) \in E$

- For each $(a, b) \in R_e$ and $i \in \{1, \ldots, p-1\}$
  - Add the equation $x_u - a = i \cdot (x_v - b)$
- Note that given an assignment to the x variables
  - If $(x_u, x_v) \notin R_e$ at most $|R_e| \le |L|^2 \le \eta(p-1)$ of the equations are satisfied
  - If $(x_u, x_v) \in R_e$
    - at least $p - 1$ of the equations are satisfied

Proof (2/3)

For each edge $e = (u, v) \in E$

- For each $(a, b) \in R_e$ and $i \in \{1, \ldots, p-1\}$
  - Add the equation $x_u - a = i \cdot (x_v - b)$
- Note that given an assignment to the x variables
  - If $(x_u, x_v) \notin R_e$ at most $|R_e| \leq |L|^2 \leq \eta(p-1)$ of the equations are satisfied
  - If $(x_u, x_v) \in R_e$
    - at least $p - 1$ of the equations are satisfied
    - at most $p - 1 + |R_e| - 1 \leq 2p - 2$ of the equations are satisfied

Proof (3/3)

- Let *Opt* be the maximum number of equations satisfied

Proof (3/3)

- Let *Opt* be the maximum number of equations satisfied
- If $\mathrm{Val}(X) = 1$, $Opt \geq |E| \cdot (p - 1)$

Proof (3/3)

- Let *Opt* be the maximum number of equations satisfied
- If $\mathrm{Val}(X) = 1$, $Opt \geq |E| \cdot (p-1)$
- If $\mathrm{Val}(X) \leq \eta$

$$Opt \leq (1-\eta)|E| \cdot \eta(p-1) + \eta|E| \cdot (2p-2)$$

Proof (3/3)

- Let *Opt* be the maximum number of equations satisfied
- If $\mathrm{Val}(X) = 1$, $Opt \geq |E| \cdot (p-1)$
- If $\mathrm{Val}(X) \leq \eta$

$$\begin{aligned} Opt &\leq (1-\eta)|E| \cdot \eta(p-1) + \eta|E| \cdot (2p-2) \\ &\leq 3\eta \cdot |E| \cdot (p-1) \end{aligned}$$

Questions!