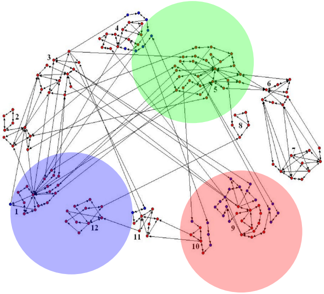
From Weak to Strong LP Gaps for all CSPs



Mrinalkanti Ghosh

Madhur Tulsiani

Max-k-CSP

Max-k-CSP

- n Boolean variables.

Max-k-CSP

- n Boolean variables.
- m constraints (each on k variables)

Max-k-CSP

- n Boolean variables.
- m constraints (each on k variables)
- Satisfy as many as possible.

Max-3-SAT

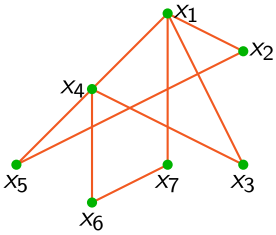
$$x_1 \vee x_{22} \vee \overline{x}_{19}$$

$$x_3 \vee \overline{x}_9 \vee x_{23}$$

$$x_5 \vee \overline{x}_7 \vee \overline{x}_9$$

$$\vdots$$

Max-Cut



$$x_1 \neq x_2$$

$$x_2 \neq x_5$$

$$x_3 \neq x_4$$

$$\vdots$$

Max-k-CSP$_q$

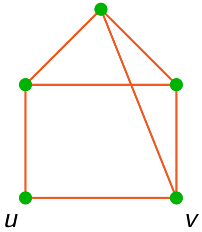
Max-k-CSP$_q$

- n variables taking values in $[q] = \{0, \ldots, q-1\}$.
- m constraints (each on k variables)
- Satisfy as many as possible.

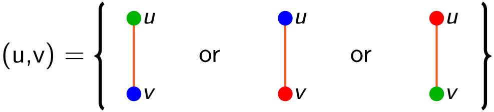
Max-k-CSP$_q$

- n variables taking values in $[q] = \{0, \ldots, q-1\}$.
- m constraints (each on k variables)
- Satisfy as many as possible.

Unique Games



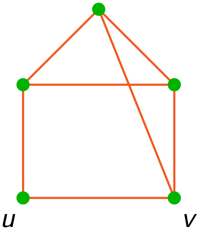
- For a graph, given:
  - Set of colors: $[q]$
  - Constraints: one for each edge $(u, v) \in E$



Max-k-CSP$_q$

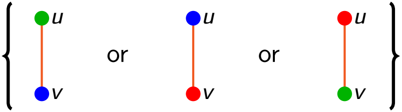
- n variables taking values in $[q] = \{0, \ldots, q-1\}$.
- m constraints (each on k variables)
- Satisfy as many as possible.

Unique Games



- For a graph, given:
  - Set of colors: $[q]$
  - Constraints: one for each edge $(u, v) \in E$

(u,v) =



- Each constraint is a bijection from $[q]$ to $[q]$. Can in fact consider difference equations

$$x_u - x_v = c_{uv} \pmod{q}$$

Max-k-CSP$_q$(f)

- Characterized by $f : [q]^k \to \{0, 1\}$.

Max-k-CSP$_q$(f)

- Characterized by $f : [q]^k \to \{0,1\}$.
- Each constraint is of the form

$$C_i \equiv f\left(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k}\right)$$

for $i_1, \ldots, i_k \in [n]$ and $b_{i,1}, \ldots, b_{i,k} \in [q]$. (addition is mod q)

Max-k-CSP$_q$(f)

- Characterized by $f : [q]^k \to \{0,1\}$.
- Each constraint is of the form

$$C_i \equiv f\left(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k}\right)$$

  for $i_1, \ldots, i_k \in [n]$ and $b_{i,1}, \ldots, b_{i,k} \in [q]$. (addition is mod q)
- Max-3-SAT: $f \equiv$ OR. Each C_i is a clause. $b_{i,1} = 1$ if x_{i_1} is negated in clause C_i.

Max-k-CSP$_q$(f)

- Characterized by $f : [q]^k \to \{0, 1\}$.

- Each constraint is of the form

$$C_i \equiv f\left(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k}\right)$$

for $i_1, \ldots, i_k \in [n]$ and $b_{i,1}, \ldots, b_{i,k} \in [q]$. (addition is mod q)

- Max-3-SAT: $f \equiv$ OR. Each C_i is a clause. $b_{i,1} = 1$ if x_{i_1} is negated in clause C_i.

- Unique Games: $f \equiv$ EQUAL. For i^{th} constraint (u, v), let $i_1 = u$, $i_2 = v$ and let $b_{i,2} - b_{i,1} = c_{uv}$

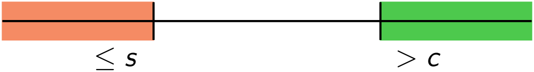
$$x_u - x_u = c_{uv} \quad \Leftrightarrow \quad x_{i_1} + b_{i,1} = x_{i_2} + b_{i,2}.$$

Approximating Max-k-CSP$_q$(f)

Relax the problem of finding maximum fraction of constraints satisfiable.

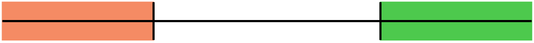
Approximating Max-k-CSP$_q$(f)

Relax the problem of finding maximum fraction of constraints satisfiable.



Approximating Max-k-CSP$_q$(f)

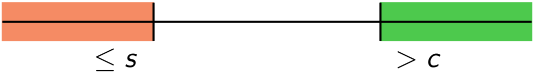
Relax the problem of finding maximum fraction of constraints satisfiable.



- Goal: Distinguish the cases $\mathrm{OPT}(\Phi) \leq s$ and $\mathrm{OPT}(\Phi) > c$.

Approximating Max-k-CSP$_q$(f)

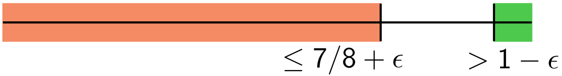
Relax the problem of finding maximum fraction of constraints satisfiable.



- Goal: Distinguish the cases $\mathrm{OPT}(\Phi) \leq s$ and $\mathrm{OPT}(\Phi) > c$.
- If for some $\gamma \leq 1$, all pairs $(\gamma \cdot c, c)$ can be solved, then can approximate within factor γ.

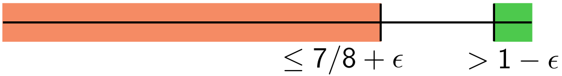
Characterizing approximability

- Max-3-SAT [Håstad 97]: For all $\epsilon > 0$, distinguishing $(7/8 + \epsilon, 1 - \epsilon)$ is NP-hard ($s < 7/8$ is trivial).

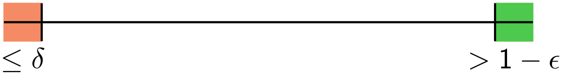


Characterizing approximability

- Max-3-SAT [Håstad 97]: For all $\epsilon > 0$, distinguishing $(7/8 + \epsilon, 1 - \epsilon)$ is NP-hard ($s < 7/8$ is trivial).



- Unique Games Conjecture [Khot 02]: For all $\delta, \epsilon > 0$, there exists q such that it is NP-hard to distinguish $(\delta, 1 - \epsilon)$ for UG with domain $[q]$.



A dichotomy assuming the UGC

- [Raghavendra 08]: For all q, for all f, if a basic SDP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, it is NP-hard to distinguish $(s + \epsilon, c - \epsilon)$ assuming the UGC.

A dichotomy assuming the UGC

- [Raghavendra 08]: For all q, for all f, if a basic SDP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, it is NP-hard to distinguish $(s + \epsilon, c - \epsilon)$ assuming the UGC.

- "All-or-nothing": Either a simple algorithm (approximately solvable in almost linear time) can distinguish (s, c) or it is NP-hard to do so.

A dichotomy assuming the UGC

- [Raghavendra 08]: For all q, for all f, if a basic SDP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, it is NP-hard to distinguish $(s + \epsilon, c - \epsilon)$ assuming the UGC.
- "All-or-nothing": Either a simple algorithm (approximately solvable in almost linear time) can distinguish (s, c) or it is NP-hard to do so.
- Equivalent to UGC (because UG is a 2-CSP).

An unconditional version for LPs [Ghosh T 17]

- For all q, for all f, if a basic LP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, no LP of any polynomial size in the Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$.

An unconditional version for LPs [Ghosh T 17]

- For all q, for all f, if a basic LP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, no LP of any polynomial size in the Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$.

- [CLRS 13], [KMR 17]: If no polysize LP in Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$ then no polysize extended formulation can distinguish $(s + 2\epsilon, c - 2\epsilon)$.

An unconditional version for LPs [Ghosh T 17]

- For all q, for all f, if a basic LP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, no LP of any polynomial size in the Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$.

- [CLRS 13], [KMR 17]: If no polysize LP in Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$ then no polysize extended formulation can distinguish $(s + 2\epsilon, c - 2\epsilon)$.

- "All-or-not-much" for LPs: If a simple (linear size) LP cannot do it, neither can any polysize LP extended formulation.

(Linear) Extended formulations

- Defined by a feasible polytope P, and a way of encoding instances Φ as a (linear) objective function w_Φ.

(Linear) Extended formulations

- Defined by a feasible polytope P, and a way of encoding instances Φ as a (linear) objective function w_Φ.

- Introduce additional variables y. Optimize over polytope $P = \{x \mid \exists y \;\; Ex + Fy = g, y \geq 0\}$.

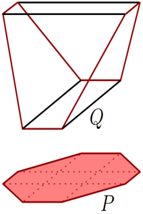


Image from [Fiorini-Rothvoss-Tiwari 2011]

(Linear) Extended formulations

- Defined by a feasible polytope P, and a way of encoding instances Φ as a (linear) objective function w_Φ.

- Introduce additional variables y. Optimize over polytope

$$P = \{x \mid \exists y \;\; Ex + Fy = g, y \geq 0\} \; .$$

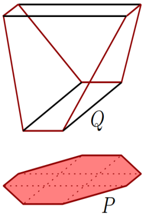


Image from [Fiorini-Rothvoss-Tiwari 2011]

(Linear) Extended formulations

- Defined by a feasible polytope P, and a way of encoding instances Φ as a (linear) objective function w_Φ.

- Introduce additional variables y. Optimize over polytope
  $P = \{x \mid \exists y \;\; Ex + Fy = g, y \geq 0\}$.

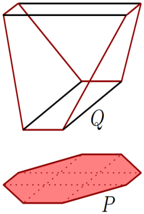


Image from [Fiorini-Rothvoss-Tiwari 2011]

Size equals #variables + #constraints.

(Linear) Extended formulations

- Defined by a feasible polytope P, and a way of encoding instances Φ as a (linear) objective function w_Φ.

- Introduce additional variables y. Optimize over polytope $P = \{x \mid \exists y \;\; Ex + Fy = g, y \geq 0\}$.

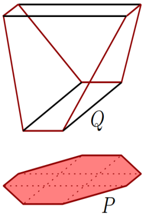


Image from [Fiorini-Rothvoss-Tiwari 2011]

Size equals #variables + #constraints.

- Optimize objective objective $\langle w_\Phi, x \rangle$ (depending on Φ) over P.

Integer Program for CSPs

Variables: $Z_{(i,b)}$ for $i \in [n]$ and $b \in [q]$

Constraints: $(Z_{(i,b)})^2 = Z_{(i,b)} \quad \forall\ i \in [n], b \in [q]$

$$\sum_{b \in [q]} Z_{(i,b)} \ = \ 1 \quad \forall\ i \in [n]$$

Integer Program for CSPs

Variables: $Z_{(i,b)}$ for $i \in [n]$ and $b \in [q]$

Constraints: $(Z_{(i,b)})^2 = Z_{(i,b)} \quad \forall\ i \in [n], b \in [q]$

$$\sum_{b \in [q]} Z_{(i,b)} \ = \ 1 \quad \forall\ i \in [n]$$

Maximize: $$\frac{1}{m} \cdot \sum_{C} \sum_{\alpha \in [q]^{S_C}} \left(\prod_{i \in S_C} Z_{(i,\alpha_i)} \right) \cdot f\left(\alpha + (b_{i,1}, \ldots, b_{i,k})\right)$$

The Sherali-Adams LP hierarchy (t levels)

The Sherali-Adams LP hierarchy (t levels)

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. Represent $\widetilde{\mathbb{E}}$ as

$$X_{(S,\alpha)} = \widetilde{\mathbb{E}}\left[\prod_{i \in S} Z_{(i,\alpha_i)}\right] \approx \text{Prob. vars in } S \text{ assigned according to } \alpha$$

$$X_{(S,\alpha)} \geq 0$$

The Sherali-Adams LP hierarchy (t levels)

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. Represent $\widetilde{\mathbb{E}}$ as

$$X_{(S,\alpha)} = \widetilde{\mathbb{E}}\left[\prod_{i \in S} Z_{(i,\alpha_i)}\right] \approx \text{Prob. vars in } S \text{ assigned according to } \alpha$$

$$X_{(S,\alpha)} \geq 0$$

The Sherali-Adams LP hierarchy (t levels)

Variables: $X_{(S,\alpha)}$ for all $|S| \le t$ and $\alpha \in [q]^S$. Represent $\widetilde{\mathbb{E}}$ as

$$X_{(S,\alpha)} = \widetilde{\mathbb{E}}\left[\prod_{i \in S} Z_{(i,\alpha_i)}\right] \approx \text{Prob. vars in } S \text{ assigned according to } \alpha$$

$$X_{(S,\alpha)} \geq 0$$

Consistency: For all $j \notin S$, $\sum_{b \in [q]} X_{(S \cup \{j\}, \alpha \circ b)} = X_{(S,\alpha)}$

$$X_{\emptyset,\emptyset} = 1$$

The Sherali-Adams LP hierarchy (t levels)

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. Represent $\widetilde{\mathbb{E}}$ as

$$X_{(S,\alpha)} = \widetilde{\mathbb{E}}\left[\prod_{i \in S} Z_{(i,\alpha_i)}\right] \approx \text{Prob. vars in } S \text{ assigned according to } \alpha$$

$$X_{(S,\alpha)} \geq 0$$

Consistency: For all $j \notin S$, $\quad \sum_{b \in [q]} X_{(S \cup \{j\}, \alpha \circ b)} = X_{(S,\alpha)}$

$$X_{\emptyset,\emptyset} = 1$$

Linear Program: For variables $X_{(S,\alpha)} \in [0,1]$ satisfying consistency

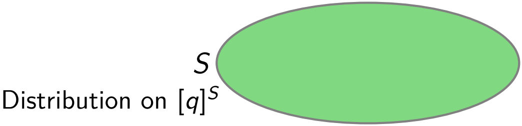
$$\text{Maximize} \quad \frac{1}{m} \cdot \sum_{C} \sum_{\alpha \in [q]^{S_C}} X_{(S_C,\alpha)} \cdot f\left(\alpha + \left(b_{i,1}, \ldots, b_{i,k}\right)\right)$$

A local distribution view

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. $X_{(S,\alpha)} \geq 0$.

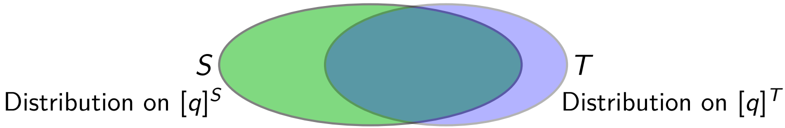
A local distribution view

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. $X_{(S,\alpha)} \geq 0$.



A local distribution view

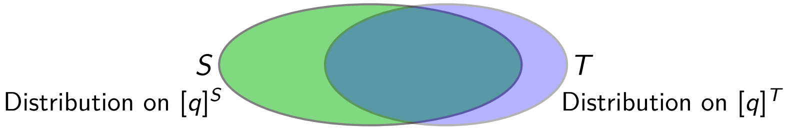
Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. $X_{(S,\alpha)} \geq 0$.



$\sum_{b \in [q]} X_{(S \cup \{j\}, \alpha \circ b)} = X_{(S,\alpha)}$

A local distribution view

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. $X_{(S,\alpha)} \geq 0$.



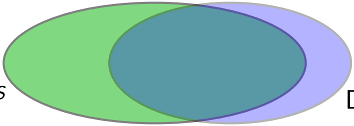
$\sum_{b \in [q]} X_{(S \cup \{j\}, \alpha \circ b)} = X_{(S,\alpha)}$

- Solution to LP defines local distributions consistent on intersections.

A local distribution view

Variables: $X_{(S,\alpha)}$ for all $|S| \leq t$ and $\alpha \in [q]^S$. $X_{(S,\alpha)} \geq 0$.

S Distribution on $[q]^S$



T Distribution on $[q]^T$

$$\sum_{b\in[q]} X_{(S\cup\{j\},\alpha\circ b)} = X_{(S,\alpha)}$$

- Solution to LP defines local distributions consistent on intersections.
- $n^{O(t)} \cdot q^t$ variables.

The basic LP

The basic LP

- Variables: $X_{(S_C,\alpha)}$ for all constraints C and $\alpha \in [q]^{S_C}$

  $X_{(S_C,\alpha)} \approx$ Probability that vars in S_C assigned according to α

  Also define $X_{(j,b)}$ for each $j \in [n]$, $b \in [q]$.

The basic LP

- Variables: $X_{(S_C,\alpha)}$ for all constraints C and $\alpha \in [q]^{S_C}$

  $X_{(S_C,\alpha)}$ $\approx$ Probability that vars in S_C assigned according to α

  Also define $X_{(j,b)}$ for each $j \in [n], b \in [q]$.

- Consistency: $\forall j \in S_{C_i}, \ \forall b \in [q], \quad \sum_{\substack{\alpha \in [q]^{S_{C_i}} \\ \alpha(j)=b}} X_{(S_{C_i},\alpha)} = X_{(j,b)}$

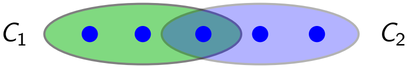
The basic LP

- Variables: $X_{(S_C,\alpha)}$ for all constraints C and $\alpha \in [q]^{S_C}$

  $X_{(S_C,\alpha)} \approx$ Probability that vars in S_C assigned according to α

  Also define $X_{(j,b)}$ for each $j \in [n], b \in [q]$.

- Consistency: $\forall j \in S_{C_i},\ \forall b \in [q], \quad \sum_{\substack{\alpha \in [q]^{S_{C_i}} \\ \alpha(j)=b}} X_{(S_{C_i},\alpha)} = X_{(j,b)}$



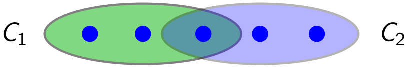
The basic LP

- Variables: $X_{(S_C,\alpha)}$ for all constraints C and $\alpha \in [q]^{S_C}$

  $X_{(S_C,\alpha)}$ $\approx$ Probability that vars in S_C assigned according to α

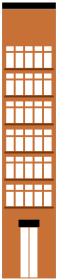
  Also define $X_{(j,b)}$ for each $j \in [n]$, $b \in [q]$.

- Consistency: $\forall j \in S_{C_i}, \ \forall b \in [q], \quad \sum_{\substack{\alpha \in [q]^{S_{C_i}} \\ \alpha(j)=b}} X_{(S_{C_i},\alpha)} = X_{(j,b)}$

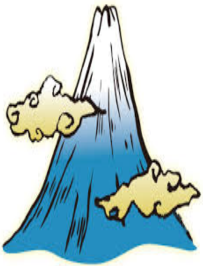


- $O(q^k \cdot m + q \cdot n)$ variables.

Inaccurate pictorial representations

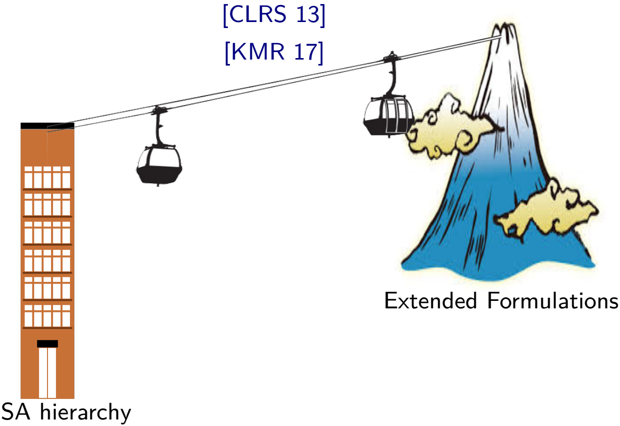


SA hierarchy



Extended Formulations

Inaccurate pictorial representations



Inaccurate pictorial representations

Max-Cut [CMM 09]

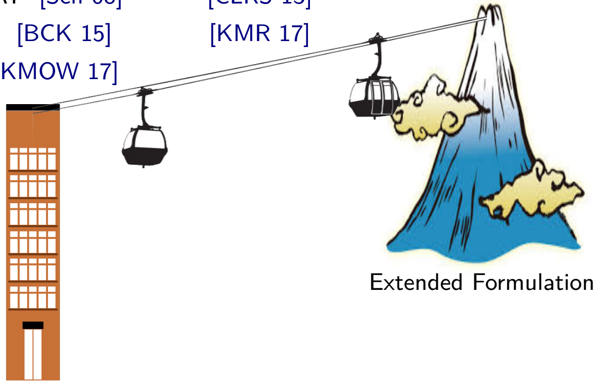
Max-3-SAT [Sch 08]

Pairwise [BCK 15]

[KMOW 17]

[CLRS 13]

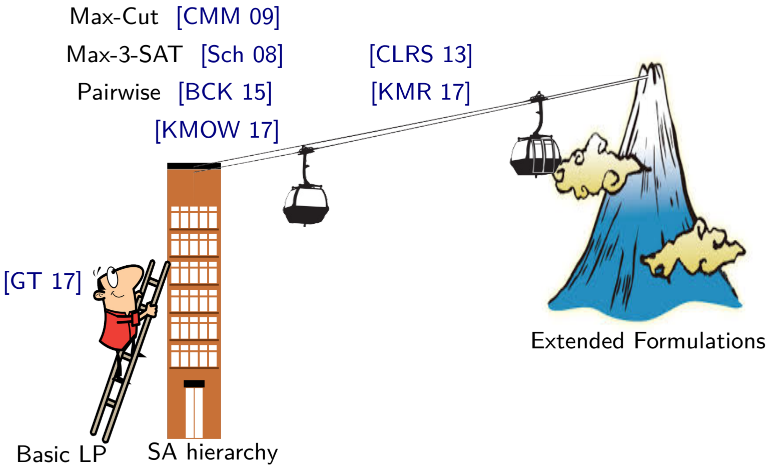
[KMR 17]



Extended Formulations

SA hierarchy

Inaccurate pictorial representations



A more precise version

- [Ghosh T 17]: For all q, for all f, if basic LP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, no LP given by $t = O_\epsilon\left(\frac{\log n}{\log \log n}\right)$ levels of the Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$.

A more precise version

- [Ghosh T 17]: For all q, for all f, if basic LP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, no LP given by $t = O_\epsilon\left(\frac{\log n}{\log\log n}\right)$ levels of the Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$.

- Using [CLRS 13, KMR 17]: For all $\epsilon > 0$, no extended formulation of size $\exp\left(O_\epsilon\left(\frac{(\log n)^2}{(\log\log n)^2}\right)\right)$ can distinguish $(s + \epsilon, c - \epsilon)$.

A more precise version

- [Ghosh T 17]: For all q, for all f, if basic LP cannot distinguish (s, c) for Max-k-CSP$_q$(f), then for all $\epsilon > 0$, no LP given by $t = O_\epsilon\left(\frac{\log n}{\log\log n}\right)$ levels of the Sherali-Adams hierarchy can distinguish $(s + \epsilon, c - \epsilon)$.

- Using [CLRS 13, KMR 17]: For all $\epsilon > 0$, no extended formulation of size $\exp\left(O_\epsilon\left(\frac{(\log n)^2}{(\log\log n)^2}\right)\right)$ can distinguish $(s + \epsilon, c - \epsilon)$.

- "Escalate" a hard instance for basic LP to a hard instance for Sherali-Adams.

What is a hard instance ($c = 1$)

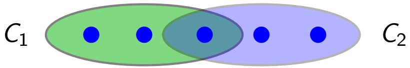
- Φ_0 is a (c, s) hard instance of basic LP, for $c = 1$ if

What is a hard instance ($c = 1$)

- Φ_0 is a (c, s) hard instance of basic LP, for $c = 1$ if
  - No assignment satisfies more than s fraction of constraints.

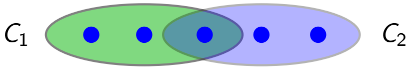
What is a hard instance ($c = 1$)

- Φ_0 is a (c, s) hard instance of basic LP, for $c = 1$ if
  - No assignment satisfies more than s fraction of constraints.
  - All local distributions on constraints are supported only on satisfying assignments.



What is a hard instance ($c = 1$)

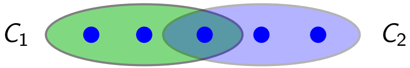
- Φ_0 is a (c, s) hard instance of basic LP, for $c = 1$ if
  - No assignment satisfies more than s fraction of constraints.
  - All local distributions on constraints are supported only on satisfying assignments.



- Using Φ_0, create a (level-t) hard instance Φ where
  - No assignment satisfies more than s fraction of constraints.

What is a hard instance ($c = 1$)

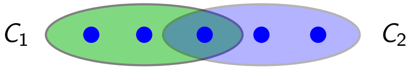
- Φ_0 is a (c, s) hard instance of basic LP, for $c = 1$ if
  - No assignment satisfies more than s fraction of constraints.
  - All local distributions on constraints are supported only on satisfying assignments.



- Using Φ_0, create a (level-t) hard instance Φ where
  - No assignment satisfies more than s fraction of constraints.
  - There exist local distributions on all subsets S, $|S| \leq t$, consistent on all intersections.

What is a hard instance ($c = 1$)

- Φ_0 is a (c, s) hard instance of basic LP, for $c = 1$ if
  - No assignment satisfies more than s fraction of constraints.
  - All local distributions on constraints are supported only on satisfying assignments.



- Using Φ_0, create a (level-t) hard instance Φ where
  - No assignment satisfies more than s fraction of constraints.
  - There exist local distributions on all subsets S, $|S| \leq t$, consistent on all intersections.
  - Distribution on S only supported on assignments satisfying (almost) all constraints in S.

Intuition for the proof

- Use hard instance (say Φ_0) for basic LP as a “template” to produce a hard instance Φ for Sherali-Adams.

Intuition for the proof

- Use hard instance (say Φ_0) for basic LP as a “template” to produce a hard instance Φ for Sherali-Adams.

- Instance Φ looks “easily satisfiable” locally.

Intuition for the proof

- Use hard instance (say Φ_0) for basic LP as a "template" to produce a hard instance Φ for Sherali-Adams.
- Instance Φ looks "easily satisfiable" locally.
- Think of instance as (hyper)graph. Each constraint adds a hyperedge. Locally like (hyper)trees.

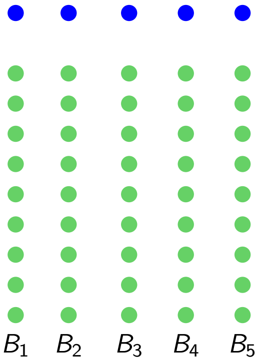
Intuition for the proof

- Use hard instance (say Φ_0) for basic LP as a “template” to produce a hard instance Φ for Sherali-Adams.
- Instance Φ looks “easily satisfiable” locally.
- Think of instance as (hyper)graph. Each constraint adds a hyperedge. Locally like (hyper)trees.
- Trees are easy.

The gap construction

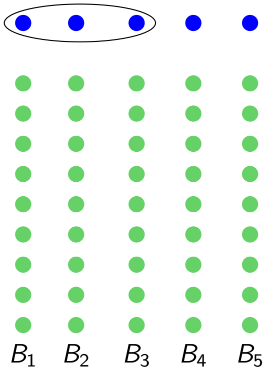
- Will use (s, c) hard instance Φ_0 for basic LP as template.

The gap construction



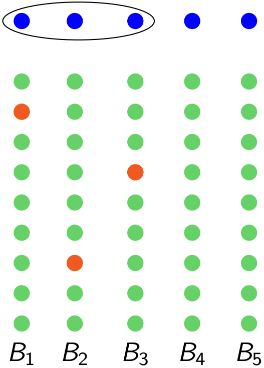
- Will use (s, c) hard instance Φ_0 for basic LP as template.
- Consider a bucket of variables B_r for every variable x_r in Φ_0. $|B_r| = n$.

The gap construction



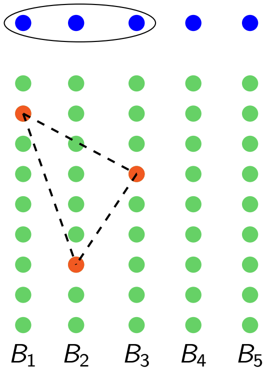
- Will use (s, c) hard instance Φ_0 for basic LP as template.
- Consider a bucket of variables B_r for every variable x_r in Φ_0. $|B_r| = n$.
- Repeat m times:
    - Sample $C \sim \Phi_0$. Let $C \equiv f(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k})$.

The gap construction



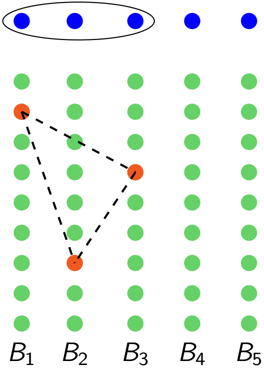
- Will use (s, c) hard instance Φ_0 for basic LP as template.
- Consider a bucket of variables B_r for every variable x_r in Φ_0. $|B_r| = n$.
- Repeat m times:
  - Sample $C \sim \Phi_0$. Let $C \equiv f(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k})$.
  - Pick j^{th} variable uniformly from bucket B_{i_j}. Let z_{i_j} be the sampled variable from this bucket.

The gap construction



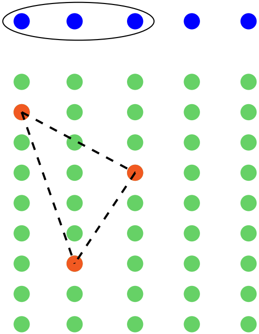
- Will use (s, c) hard instance Φ_0 for basic LP as template.
- Consider a bucket of variables B_r for every variable x_r in Φ_0. $|B_r| = n$.
- Repeat m times:
  - Sample $C \sim \Phi_0$. Let $C \equiv f(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k})$.
  - Pick j^{th} variable uniformly from bucket B_{i_j}. Let z_{i_j} be the sampled variable from this bucket.
  - Include constraint $f(z_{i_1} + b_{i,1}, \ldots, z_{i_k} + b_{i,k})$.

The gap construction



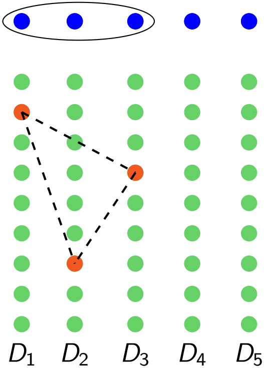
- Will use (s, c) hard instance Φ_0 for basic LP as template.
- Consider a bucket of variables B_r for every variable x_r in Φ_0. $|B_r| = n$.
- Repeat m times:
  - Sample $C \sim \Phi_0$. Let $C \equiv f(x_{i_1} + b_{i,1}, \ldots, x_{i_k} + b_{i,k})$.
  - Pick j^{th} variable uniformly from bucket B_{i_j}. Let z_{i_j} be the sampled variable from this bucket.
  - Include constraint $f(z_{i_1} + b_{i,1}, \ldots, z_{i_k} + b_{i,k})$.
- Similar constructions used by [GL 15], [KTW 14]

Bounding OPT(Φ)



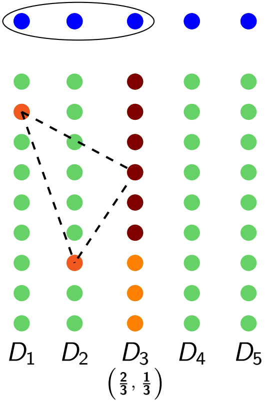
- Fix an assignment σ to all vars in new instance Φ

Bounding OPT(Φ)



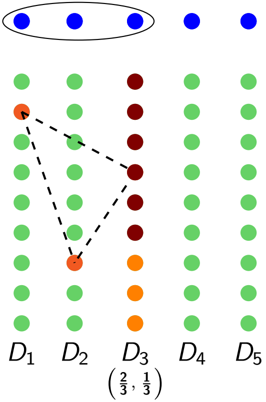
- Fix an assignment σ to all vars in new instance Φ
- Let D_r be the empirical distribution on $[q]$ for variables in B_r.

Bounding OPT(Φ)



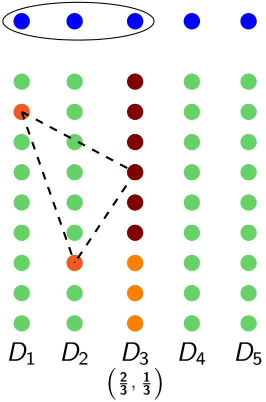
- Fix an assignment σ to all vars in new instance Φ
- Let D_r be the empirical distribution on $[q]$ for variables in B_r.

Bounding OPT(Φ)



- Fix an assignment σ to all vars in new instance Φ
- Let D_r be the empirical distribution on $[q]$ for variables in B_r.
- Let x_r be a var in constraint $C \in \Phi_0$. A random copy of C sees a value for this variable independently distributed with D_r.

Bounding OPT(Φ)

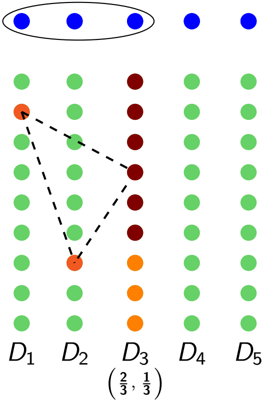


- Fix an assignment σ to all vars in new instance Φ
- Let D_r be the empirical distribution on $[q]$ for variables in B_r.
- Let x_r be a var in constraint $C \in \Phi_0$. A random copy of C sees a value for this variable independently distributed with D_r.
- For a fixed σ,

$$\mathbb{E}_{\Phi}\left[\text{Fraction of sat. constraints in } \Phi\right]$$

equals fraction satisfied in Φ_0 by rounding each x_r independently from D_r ($\leq s$).

Bounding OPT(Φ)

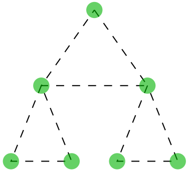


- Fix an assignment σ to all vars in new instance Φ
- Let D_r be the empirical distribution on $[q]$ for variables in B_r.
- Let x_r be a var in constraint $C \in \Phi_0$. A random copy of C sees a value for this variable independently distributed with D_r.
- For a fixed σ,

$$\mathbb{E}_\Phi \left[\text{Fraction of sat. constraints in } \Phi\right]$$

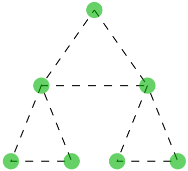
equals fraction satisfied in Φ_0 by rounding each x_r independently from D_r ($\leq s$).
- Concentration and union bound.

Propagation on trees



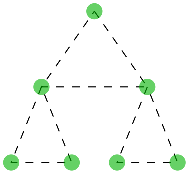
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.

Propagation on trees



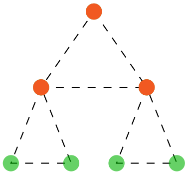
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).

Propagation on trees



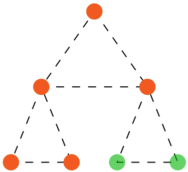
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).

Propagation on trees



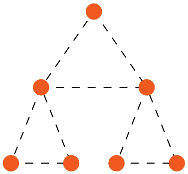
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).

Propagation on trees



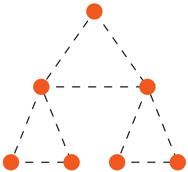
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).

Propagation on trees



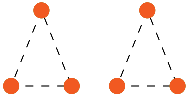
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).

Propagation on trees



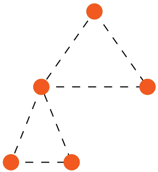
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).
- Does not depend on choice of root.

Propagation on trees



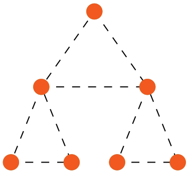
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).
- Does not depend on choice of root.
- May not be consistent between tree and disconnected sub-forest.

Propagation on trees



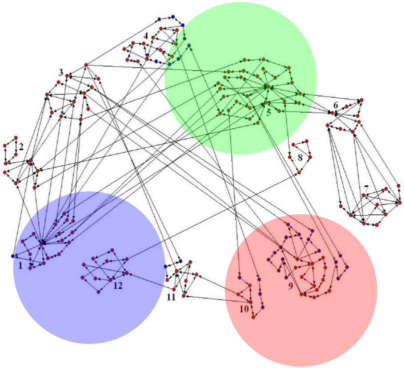
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).
- Does not depend on choice of root.
- May not be consistent between tree and disconnected sub-forest.
- Is consistent on a subtree.

Propagation on trees



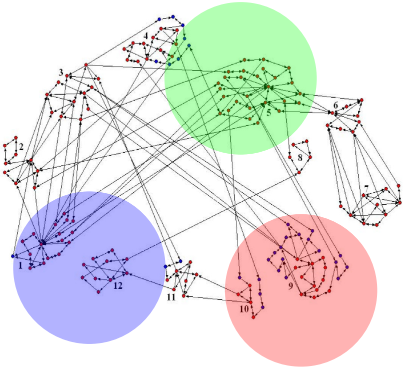
- Random hypergraphs have no cycles of size $O(\log n)$. Locally like trees.
- Each hyperedge e in a tree comes from a constraint in Φ_0. Comes with a given distribution on e (from basic LP).
- Propagate to child conditioned on parent. Can be done by consistency on variables (vertices).
- Does not depend on choice of root.
- May not be consistent between tree and disconnected sub-forest.
- Is consistent on a subtree.

Breaking up the graph



- Idea: Given set $S \subseteq V$, break S into low-diameter components. Connect all paths in each component - always a tree, never a forest.

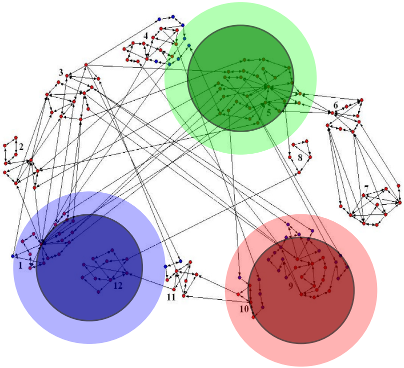
Breaking up the graph



- Idea: Given set $S \subseteq V$, break S into low-diameter components. Connect all paths in each component - always a tree, never a forest.

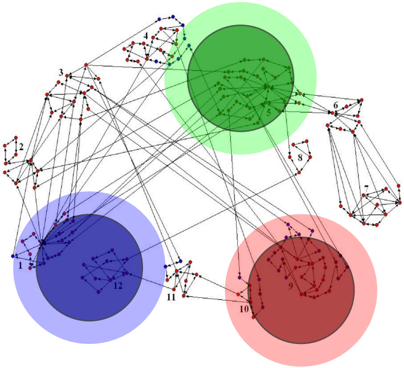
- Propagate on each component tree.

Breaking up the graph



- Idea: Given set $S \subseteq V$, break S into low-diameter components. Connect all paths in each component - always a tree, never a forest.
- Propagate on each component tree.
- If $T \subset S$, distribution on components of T induced by S should be same as obtained by partitioning T.

Breaking up the graph



- Idea: Given set $S \subseteq V$, break S into low-diameter components. Connect all paths in each component - always a tree, never a forest.

- Propagate on each component tree.

- If $T \subset S$, distribution on components of T induced by S should be same as obtained by partitioning T.

- Cut only few edges.

Subset consistent partioning schemes

Subset consistent partioning schemes

- [CMM 07]: Define a metric ρ on random (hyper)graph H

$$(\rho(u,v))^2 \approx 1-(1-\mu)^{d_H(u,v)}$$

ρ embeds in ℓ_2 on small sets S (for small enough μ).

Subset consistent partioning schemes

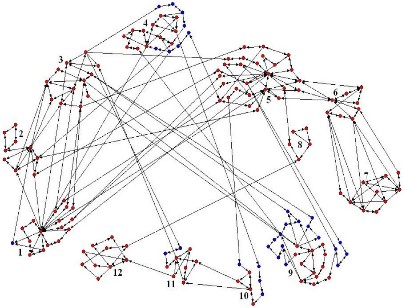
- [CMM 07]: Define a metric ρ on random (hyper)graph H

  $$(\rho(u,v))^2 \approx 1-(1-\mu)^{d_H(u,v)}$$

  ρ embeds in ℓ_2 on small sets S (for small enough μ).

- [CCGGP 98]: Low-diameter decomposition of ℓ_2 embedding.

Subset consistent partioning schemes



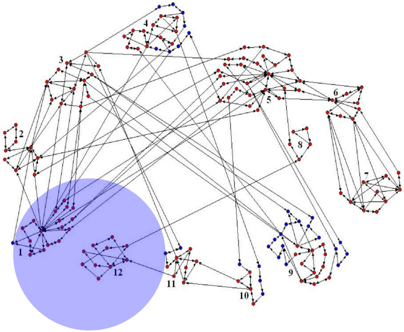
- [CMM 07]: Define a metric ρ on random (hyper)graph H

$$(\rho(u,v))^2 \approx 1-(1-\mu)^{d_H(u,v)}$$

ρ embeds in ℓ_2 on small sets S (for small enough μ).

- [CCGGP 98]: Low-diameter decomposition of ℓ_2 embedding.

Subset consistent partioning schemes



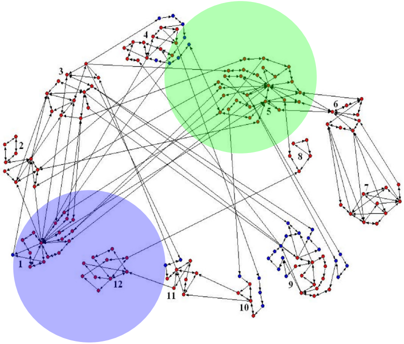
- [CMM 07]: Define a metric ρ on random (hyper)graph H

  $$(\rho(u,v))^2 \approx 1 - (1-\mu)^{d_H(u,v)}$$

  ρ embeds in ℓ_2 on small sets S (for small enough μ).

- [CCGGP 98]: Low-diameter decomposition of ℓ_2 embedding.

Subset consistent partioning schemes



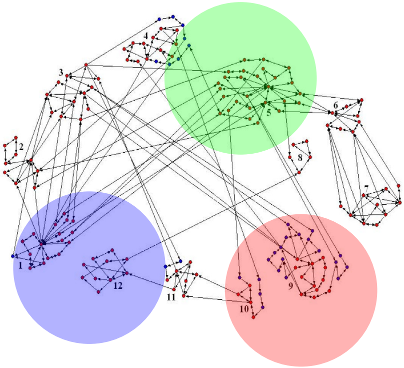
- [CMM 07]: Define a metric ρ on random (hyper)graph H

$$(\rho(u,v))^2 \approx 1-(1-\mu)^{d_H(u,v)}$$

  ρ embeds in ℓ_2 on small sets S (for small enough μ).

- [CCGGP 98]: Low-diameter decomposition of ℓ_2 embedding.

Subset consistent partioning schemes



- [CMM 07]: Define a metric ρ on random (hyper)graph H

$$(\rho(u,v))^2 \approx 1 - (1-\mu)^{d_H(u,v)}$$

  ρ embeds in ℓ_2 on small sets S (for small enough μ).

- [CCGGP 98]: Low-diameter decomposition of ℓ_2 embedding.

- Easy to check partitioning is consistent on subsets (ℓ_2 distances determine configuration).

The dimensionality problem

- Low-diameter decomposition in $\mathbb{R}^d$ cuts each edge with probability $O(\sqrt{\mu \cdot d})$.

The dimensionality problem

- Low-diameter decomposition in $\mathbb{R}^d$ cuts each edge with probability $O(\sqrt{\mu \cdot d})$.

- For $|S| = t$, ℓ_2 embedding is in $\mathbb{R}^t$. Probability of cutting an edge is $O(\sqrt{\mu \cdot t})$. Limits t to $O(\frac{\log n}{\log \log n})$.

The dimensionality problem

- Low-diameter decomposition in $\mathbb{R}^d$ cuts each edge with probability $O(\sqrt{\mu \cdot d})$.

- For $|S| = t$, ℓ_2 embedding is in $\mathbb{R}^t$. Probability of cutting an edge is $O(\sqrt{\mu \cdot t})$. Limits t to $O(\frac{\log n}{\log \log n})$.

- [JL 84]: Random Gaussian projection in $O(\log t)$ dimensions approximately preserves all distances with high probability.

The dimensionality problem

- Low-diameter decomposition in $\mathbb{R}^d$ cuts each edge with probability $O(\sqrt{\mu \cdot d})$.
- For $|S| = t$, ℓ_2 embedding is in $\mathbb{R}^t$. Probability of cutting an edge is $O(\sqrt{\mu \cdot t})$. Limits t to $O(\frac{\log n}{\log \log n})$.
- [JL 84]: Random Gaussian projection in $O(\log t)$ dimensions approximately preserves all distances with high probability.
- For sets S and T, can one consistently discard bad Gaussian projections?

Open Problems

- Extend the result to $n^{\Omega(1)}$ levels of the SA hierarchy. Will give a size bound of $\exp(n^{\Omega(1)})$ on extended formulation size using [KMR17].

Open Problems

- Extend the result to $n^{\Omega(1)}$ levels of the SA hierarchy. Will give a size bound of $\exp(n^{\Omega(1)})$ on extended formulation size using [KMR17].

- "All-or-nothing" for Sum-of-Squares SDP hierarchy. Would give strong evidence for the UGC. Even results for specific CSPs would be interesting ($k \geq 3$?).

Open Problems

- Extend the result to $n^{\Omega(1)}$ levels of the SA hierarchy. Will give a size bound of $\exp(n^{\Omega(1)})$ on extended formulation size using [KMR17].

- "All-or-nothing" for Sum-of-Squares SDP hierarchy. Would give strong evidence for the UGC. Even results for specific CSPs would be interesting ($k \geq 3$?).

- Can one avoid loss of ϵ in c when $c = 1$ (relevant for refutation)? Exact refutation addressed by [TZ 16].

Thank You

Questions?