Optimal Constant-Time Approximation Algorithms and (Unconditional) Inapproximability Results for Every Bounded-Degree CSP

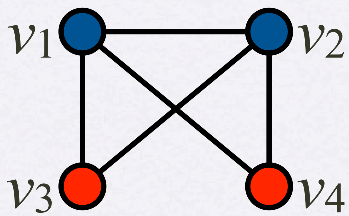
Yuichi Yoshida
Kyoto Univ. & Preferred Infrastructure

Polynomial-Time Approximation for Max CSP

Max CSP
(Constraint Satisfaction Problem)

- Given variables and constraints on them. Satisfy constraints as many as possible by assigning values to variables.

- Ex.: Max Cut (, Max *k*-SAT, Max E3LIN2)
  Input I: $(v_1 \oplus v_2), (v_1 \oplus v_3), (v_1 \oplus v_4), (v_2 \oplus v_3), (v_2 \oplus v_4)$



$\beta = (0,0,1,1)$
$\mathrm{opt}(I) = \mathrm{val}(I, \beta) = 4$

- In this talk, we mainly deal with Max Cut. But, we can use the same machinery to "any" CSP.

constant domain size
constant arity

Known Results for Poly-Time Approximation

- Since Max CSP is NP-Hard in general, approximation have been considered.

- Approximation by SDP (semidefinite programmings)
- Hardness by PCP/Unique Games Conjecture

CSP	SDP	UG-Hard
Max k-SAT	0.787 or more [ABZ06]	?
Max Cut	0.878...[GW95]	0.878...+ε[KKMO04]
Max Dicut	0.874 or more[LLZ06]	?
Max k-CSP	$\mathrm{poly}(k)/2^k$[CMM09]	$\mathrm{poly}(k)/2^k$[ST06]

Known Results for Poly-time Approximation

[Rag08] (informal)
For every CSP Λ, under Unique Games Conjecture, "BasicSDP" + a certain rounding is the best possible poly-time approximation algorithm.

CSP	SDP	UG-Hard
Max *k*-SAT	coincides (up to ε)	
Max Cut	coincides (up to ε) ≈ 0.878...	
Max Dicut	coincides (up to ε)	
Max *k*-CSP	coincides (up to ε)	

Constant-Time Approximation for Max CSP

Const-Time Approximation for Max CSP

- We want faster approximation algorithms! (O(1) time)

- A value x is an (α, ε)-approximation to x^* if:

$$\alpha x^* - \varepsilon n \leqq x \leqq x^*$$

- An assignment β is (α, ε)-approximate assignment for an input I if val(I, β) is (α, ε)-approximation to opt(I)

Bounded-Degree Model

- It takes $\Omega(n)$ to read the whole input. Thus, we read it through an oracle.

- An input $I = (V, \boldsymbol{P})$ is given as an oracle O_I: $V \times [t] \rightarrow \boldsymbol{P}$ (t = degree bound).

$$O_I(v, i) = \text{the } i\text{-th constraint incident to } v.$$

- Query complexity: # of accesses to the oracle.

Can we show something similar to [Rag08] on (α, ε)-approximation?

Our Results

- (informal) For every CSP Λ, unconditionally, "BasicLP"+ a certain rounding is the best possible constant-time approximation algorithm.

CSP	O(1) queries (via LP)	need $\Omega(\sqrt{n})$ queries
Max k-SAT	coincides (up to ε) ≈ 0.75	
Max Cut	coincides (up to ε) ≈ 0.5	
Max Dicut	coincides (up to ε) ≈ 0.5	
Max k-CSP	coincides (up to ε) $\approx 2/2^k$	

Our Results in Detail

- lp(I): the optimal value of BasicLP for an input I.
- Integrality Gap: $$\alpha_\Lambda = \inf_{I \in \Lambda} \frac{\mathrm{opt}(I)}{\mathrm{lp}(I)}$$

[Theorem] For every CSP Λ:

$\forall \varepsilon > 0$, there exists a constant-time $(\alpha_\Lambda - \varepsilon, \varepsilon)$-approximation algorithm.

$\forall \varepsilon > 0$, $\exists \delta > 0$, any algorithm that outputs $(\alpha_\Lambda + \varepsilon, \delta)$-approximation to opt($I$) with prob $\geq 2/3$ requires $\Omega(\sqrt{n})$ queries.

Query complexity: $\exp(\exp(\mathrm{poly}(qst/\varepsilon)))$

Our Results in Detail

- lp(I): the optimal value of BasicLP for an input I.
- Integrality Gap: $$\alpha_\Lambda = \inf_{I \in \Lambda} \frac{\mathrm{opt}(I)}{\mathrm{lp}(I)}$$

[Theorem] For every CSP Λ:

∀ε>0, there exists a constant-time (α_Λ-ε, ε)-approximation algorithm.

With prob ≥ 2/3, gives an oracle access to some (α_Λ-ε,ε)-approx assignment β. Once we succeed, we can compute β_v in constant time for each v.

Query complexity: exp(exp(poly(qst/ε)))

Our Results in Detail

- lp(I): the optimal value of BasicLP for an input I.
- Integrality Gap: $$\alpha_\Lambda = \inf_{I \in \Lambda} \frac{\mathrm{opt}(I)}{\mathrm{lp}(I)}$$

[Theorem] For every CSP Λ:

$\forall\varepsilon>0$, there exists a constant-time $(\alpha_\Lambda-\varepsilon, \varepsilon)$-approximation algorithm.

$\forall\varepsilon>0$, $\exists\delta>0$, any algorithm that outputs $(\alpha_\Lambda+\varepsilon, \delta)$-approximation to opt($I$) with prob $\geq 2/3$ requires $\Omega(\sqrt{n})$ queries.

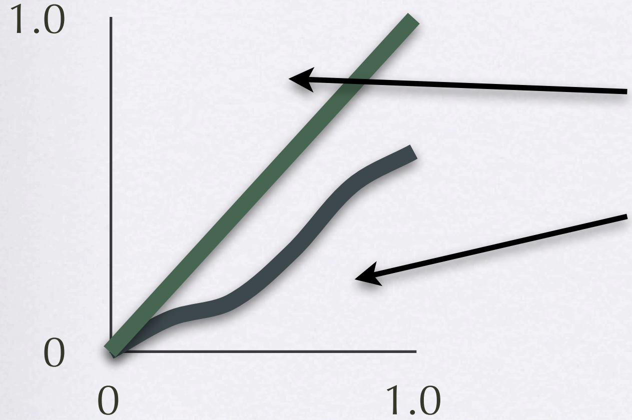
Query complexity: $\exp(\exp(\mathrm{poly}(qst/\varepsilon)))$

Upper Bound in More Detail

- Integrality gap curve: $\mathcal{S}_\Lambda(c) = \inf_{\substack{I \in \Lambda,\\ \mathrm{lp}(I) \geq cm}} \frac{\mathrm{opt}(\mathcal{I})}{m}$

- For every CSP Λ, $\varepsilon > 0$, a constant-time algorithm exists satisfying the following:

  For an input I with $\mathrm{lp}(I) = cm$ ($c \in (0,1]$), with prob $\geq$ 2/3, it gives an oracle access to β such that $S_\Lambda(c\text{-}\varepsilon)m\text{-}\varepsilon n \leqq \mathrm{val}(I, \beta) \leqq \mathrm{opt}(I)$.



$\mathrm{opt}(I) \geqq cm$

$\mathrm{opt}(I) \leqq S_\Lambda(c\text{-}\varepsilon)m\text{-}\varepsilon n$

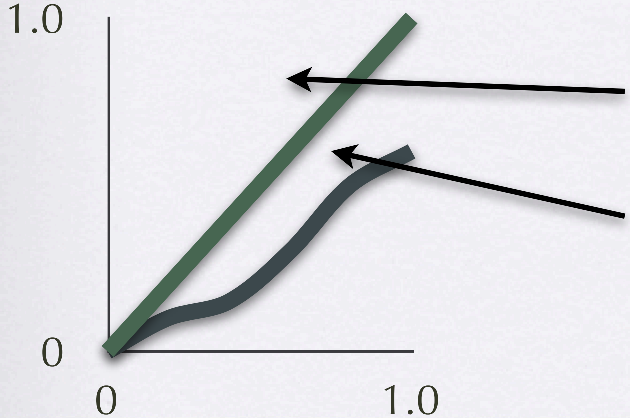
are distinguishable in const time.

Lower Bounds in More Detail

- Integrality gap curve: $\mathcal{S}_\Lambda(c) = \inf_{\substack{I \in \Lambda, \\ \mathrm{lp}(I) \geq cm}} \frac{\mathrm{opt}(\mathcal{I})}{m}$

- For every CSP Λ, $c \in [0, 1]$, $\varepsilon > 0$, any algorithm satisfying the following requires $\Omega(\sqrt{n})$ queries:

  For an input I with $\mathrm{opt}(I) = cm$, with prob $\geq 2/3$, it outputs a value x such that $(S_\Lambda(c)+\varepsilon)m \leqq x \leqq \mathrm{opt}(I)$.



$\mathrm{opt}(I) \geqq cm$

$\mathrm{opt}(I) \leqq (S_\Lambda(c)+\varepsilon)m$

are indistinguishable in const time.

Comparison to [Rag08]

This work	[Rag08]
For every CSP, BasicLP is the best algorithm.	For every CSP, BasicSDP is the best algorithm.
Unconditional	Assuming UGC
Lower bounds hold for satisfiable instances	Lower bounds do not hold for satisfiable instances

Property Testing

- An input I is ε-far from satisfiability: we need to remove $\varepsilon t n$ constraints to make I satisfiable.
- CSP Λ is testable: we can decide with prob ≥ 2/3 whether an input of CSP Λ is satisfiable or ε-far.

- If $\mathrm{lp}(I) = m$ implies $\mathrm{opt}(I) = m$, then CSP Λ is testable in constant time.

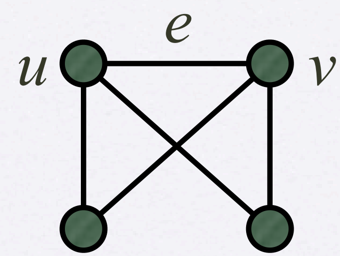
  (If integrality gap curve is continuous at $c = 1$.)

- If not, testing CSP Λ requires $\Omega(\sqrt{n})$ queries.

Proof sketch:
Lower bounds

$(\alpha_\Lambda+\varepsilon, \delta)$-approximation needs $\Omega(\sqrt{n})$ queries.

BasicLP for Max Cut

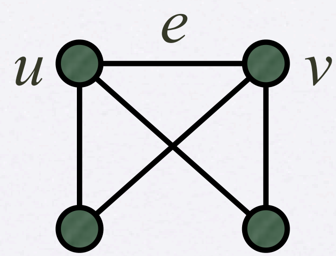


- Consider the following IP.
- $x_{v,i}$: indicating v has a value $i \in \{0,1\}$
- $\mu_{e,\beta}$: indicating e has an assignment $\beta \in \{0,1\}^2$

$$
\begin{array}{lll}
\max & \Sigma_e w_e(\mu_{e,01} + \mu_{e,10}) & \\
\text{s.t.} & x_{v,0} + x_{v,1} = 1 & \forall\, v \\
& \mu_{e,00} + \mu_{e,01} = x_{v,0} & \forall\, e = (v, u) \\
& \mu_{e,10} + \mu_{e,11} = x_{v,1} & \forall\, e = (v, u) \\
& x_{v,i} \in \{0,1\} & \forall\, v, i \\
& \mu_{e,\beta} \in \{0,1\} & \forall\, e, \beta
\end{array}
$$

BasicLP for Max Cut

- Relax the IP to LP.
- x_v: probability distribution of value of v.
- μ_e: probability distribution of assignment to e.



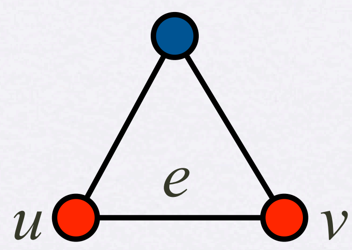
$$\max \ \Sigma_e w_e(\mu_{e,01} + \mu_{e,10})$$

$$\begin{aligned}
\text{s.t.} \quad & x_{v,0} + x_{v,1} = 1 && \forall\, v \\
& \mu_{e,00} + \mu_{e,01} = x_{v,0} && \forall\, e = (v, u) \\
& \mu_{e,10} + \mu_{e,11} = x_{v,1} && \forall\, e = (v, u) \\
& x_{v,i} \geqq 0 && \forall\, v, i \\
& \mu_{e,\beta} \geqq 0 && \forall\, e, \beta
\end{aligned}$$

Proof Strategy

- Choose I s.t. $\mathrm{lp}(I) = cm$, $\mathrm{opt}(I) \approx \alpha_\Lambda cm$.
- Create two distributions of inputs using I:
- D_I^{opt}: generates J s.t. $\mathrm{opt}(J) \leqq (\alpha_\Lambda c+\varepsilon)m$.
- D_I^{lp}: generates J s.t. $\mathrm{opt}(J) \geqq cm$.

I



$\mathrm{opt}(I) = 2 / 3$, $\mathrm{lp}(I) = 1$

$\mu_{e,00}=\mu_{e,11}=0$
$\mu_{e,01}=\mu_{e,10}=1/2$

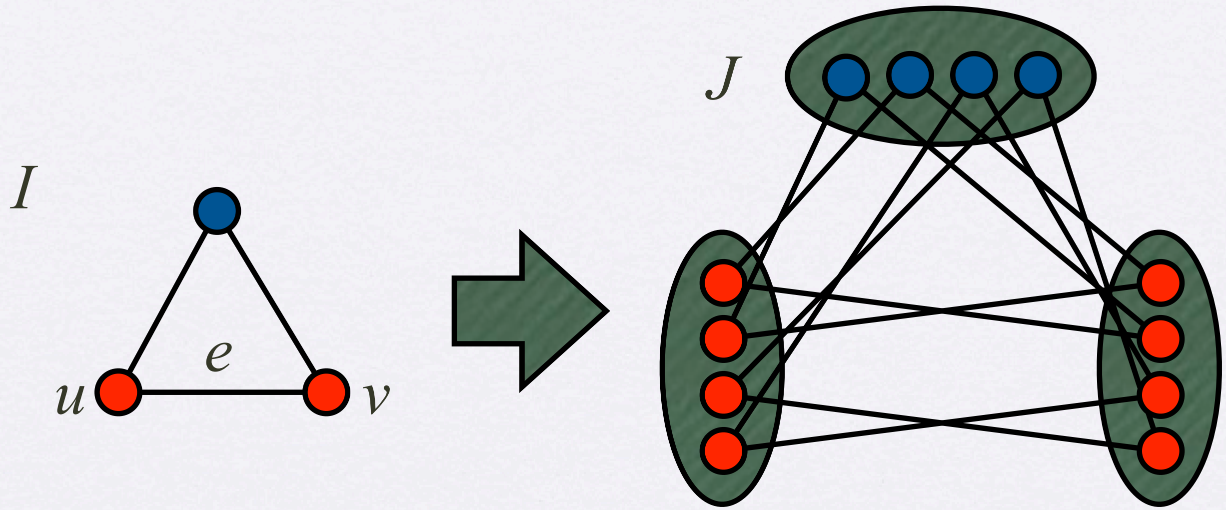
Yao's minimax principle

- Let A be a deterministic algorithm supposed to output:
  - "Yes" if J is generated by D_I^{opt}
  - "No" if J is generated by D_I^{lp}
- Let G_I^{opt}, G_I^{lp} be the distribution of subgraphs seen by A running on D_I^{opt}, D_I^{lp}, respectively.
- It suffices to show that G_I^{opt} and G_I^{lp} are "close" when # of queries is $o(\sqrt{n})$

Construction of D_I^{opt}

1. Make clusters by duplicating each vertex of *I*.
2. Make an expander for each edge of *I*.

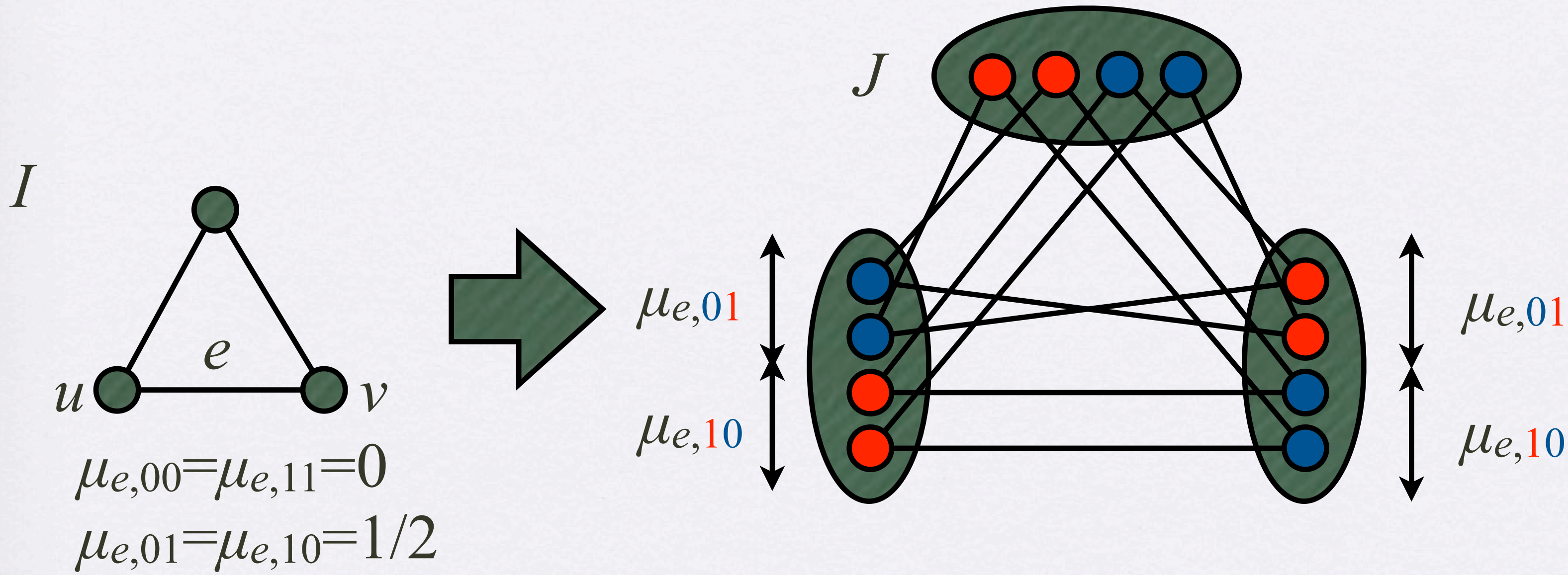
- The optimal assignment for *J* and *I* are similar.
- With prob 1-o(1), opt(*J*) ≦ $(\alpha_\Lambda c+\varepsilon)m$



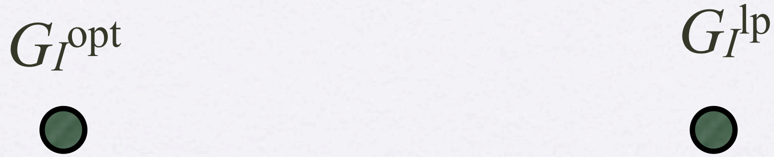
Construction of D_I^{lp}

1. Make clusters by duplicating each vertex of *I*.
2. Make an expander for each edge of *I* using μ.

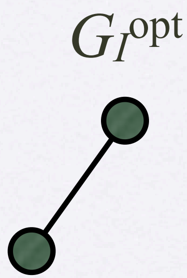
- The optimal assignment for *J* can be made from μ.
- opt(*J*) ≧ *cm*

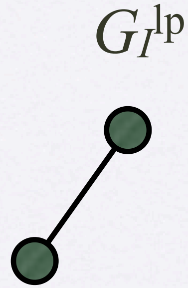


- As long as we do not find a cycle, G_I^{opt} and G_I^{lp} behaves identically.

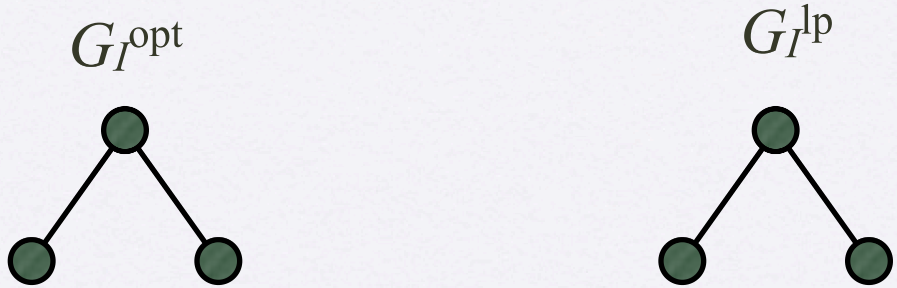


- As long as we do not find a cycle, G_I^{opt} and G_I^{lp} behaves identically.



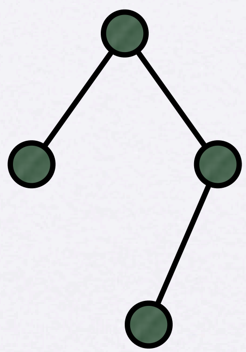


- As long as we do not find a cycle, G_I^{opt} and G_I^{lp} behaves identically.

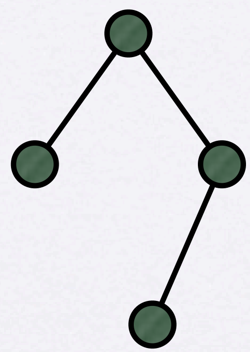


- As long as we do not find a cycle, G_I^{opt} and G_I^{lp} behaves identically.

G_I^{opt}

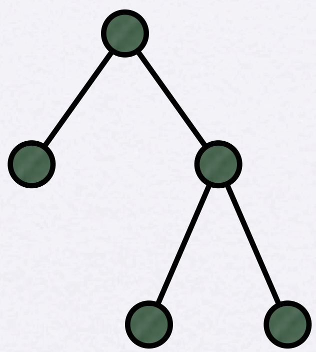


G_I^{lp}

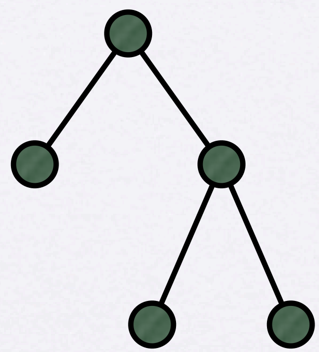


- As long as we do not find a cycle, G_I^{opt} and G_I^{lp} behaves identically.

G_I^{opt}

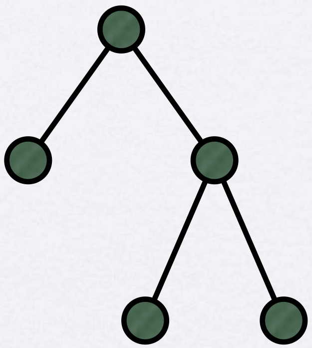


G_I^{lp}

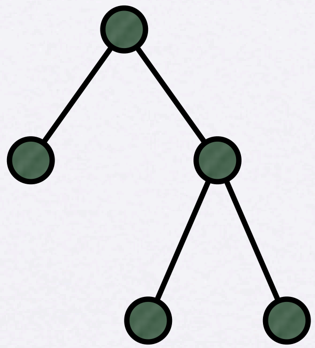


- As long as we do not find a cycle, G_I^{opt} and G_I^{lp} behaves identically.

G_I^{opt}



G_I^{lp}



For both distributions, with $o(\sqrt{n})$ queries, with high probability, we do not find a cycle
$\Rightarrow \Omega(\sqrt{n})$ bound

Proof Sketch: Upper bounds

(α_Λ-ε, ε)-approximation algorithms

Sketch of Our Algorithm

- The following (poly-time) algorithm is an optimal rounding (slight modification of [RS09]):

1. Contract vertices of I having similar LP values.
   ➡Get instance I' with constant number of vertices.
2. Compute the optimal assignment β for I' by exhaustive search.

3. Output β as an assignment for I.

Sketch of Our Algorithm

- The following (poly-time) algorithm is an optimal rounding (slight modification of [RS09]):

1. Contract vertices of I having similar LP values.
   ➡Get instance I' with constant number of vertices.
2. Compute the optimal assignment β for I' by exhaustive search.

3. Output β as an assignment for I.

Simulate in constant time!

Sketch of Our Algorithm

- For each assignment β for *I'*:
  - Sample $s = O(1)$ edges in *I*.
  - Map each edge into an edge in *I'*.
  - $\widetilde{\mathrm{val}}(I', \beta)$:= (# of edges satisfied in *I'* by β) * m / s.
- Take β that attains the maximum $\widetilde{\mathrm{val}}(I', \beta)$

- To compute LP values, we use a distributed algorithm for LP [KMW06]

Future Works: Approximation

- To what extent can we approximate with Ω(√n) queries?
    - Approimability/Inapproximability using Lovasz-Schrijver or Sherali-Adams hierarchies?
    - For Max Cut/Unique Games, can we do something with SDP, random walks or spectral technique?

Future Works: Testing

- For which CSP, $\mathrm{lp}(I) = m$ implies $\mathrm{opt}(I) = m$?
  - Ex. Horn SAT
- CSP Λ has width k if it can be solved by a certain propagation algorithm that considers a set of k variables at a time.
- Fact: Every CSP has width 1, 3 or ∞.

Conjecture:
CSP Λ is testable if and only if Λ has width 1.

Future Works: Testing

- Known results:
  - Horn SAT: $\Theta(1)$ queries [YK10]
  - 2-colorability: $\tilde{\Theta}(\sqrt{n})$ queries [GR02]
  - System of linear equations: $\Theta(n)$ queries [BOT02]

- Can we test 2SAT with $\tilde{O}(\sqrt{n})$ queries?

- The following trichotomy holds?
  - $\Theta(1)$ queries $\Leftrightarrow$ width 1
  - $\tilde{\Theta}(\sqrt{n})$ queries $\Leftrightarrow$ width 3
  - $\Theta(n)$ queries $\Leftrightarrow$ width ∞