Improved NP-inapproximability for 2-variable linear equations

Johan Håstad
KTH

Sangxia Huang
KTH

Rajsekar Manokaran
KTH

Ryan O'Donnell
CMU

John Wright
CMU

2Lin

$$x_1 = x_5$$

$$x_{10} = -x_3$$

$$x_{61} = -x_{24}$$

...

$$x_{48} = -x_5$$

(x_i = -1,1)

2Lin

2Lin(**2**) ∈ 2Lin(**q**) ≈ **UniqueGames**

$$x_1 = x_5$$
$$x_{10} = -x_3$$
$$x_{61} = -x_{24}$$
$$\dots$$
$$x_{48} = -x_5$$

(x_i = -1,1)

2Lin

$2\text{Lin}(\mathbf{2}) \in 2\text{Lin}(\mathbf{q}) \approx$ **UniqueGames**

$x_1 = x_5$

$x_{10} = -x_3$

$x_{61} = -x_{24}$

...

$x_{48} = -x_5$

(Actually, simplest case of **UG**)

(x_i = -1,1)

2Lin

$x_1 = x_5$
$x_{10} = -x_3$
$x_{61} = -x_{24}$
...
$x_{48} = -x_5$

(x_i = -1,1)

2Lin(**2**) ∈ 2Lin(**q**) ≈ **UniqueGames**

(Actually, simplest case of **UG**)

Folklore wisdom: get 2Lin(**2**) right and 2Lin(**q**) will follow.

Known results

Suppose val($\mathcal{I}$) = **α**. Can we guarantee a solution of value **C*α**?

Known results

Suppose val($\mathcal{I}$) = **α**. Can we guarantee
a solution of value **C*α**?

[GW]: .878-approx algorithm

Known results

Suppose val(I) = **α**. Can we guarantee
a solution of value **C*α**?

[GW]: .878-approx algorithm

[KKMO]+**[MOO]**: (.878+**ε**)-approx **UG**-hard

Known results

Suppose val($\mathcal{I}$) = **α**. Can we guarantee a solution of value **C*α**?

[GW]: .878-approx algorithm

[KKMO]+**[MOO]**: (.878+**ε**)-approx **UG**-hard

[Håstad]+**[TSSW]**: $^{16}/_{17}$ ≈ .941-approx **NP**-hard

Known results

Suppose val($\mathcal{I}$) = **α**. Can we guarantee

a solution of value **C*α**?

[GW]: .878-approx algorithm

[KKMO]+**[MOO]**: (.878+**ε**)-approx **UG**-hard

[Håstad]+**[TSSW]**: $^{16}/_{17} \approx .941$-approx **NP**-hard

seems we're close, right?

A different perspective...

Suppose val($\mathcal{I}$) = (1 - **ε**).

Can we guarantee a solution of value (1 - **C*ε**)?

A different perspective...

Suppose val($\mathcal{I}$) = (1 - **ε**).

Can we guarantee a solution of value (1 - **C*ε**)?

Def: Such an algo. gives an (**ε**, **C*ε**)-approx.

A different perspective...

Suppose val($\mathcal{I}$) = (1 - ε).

Can we guarantee a solution of value (1 - $f(\varepsilon)$)?

Def: Such an algo. gives an (ε, $f(\varepsilon)$)-approx.

A different perspective...

Suppose val($\mathcal{I}$) = (1 - **ε**).

Can we guarantee a solution of value (1 - **f(ε)**)?

Def: Such an algo. gives an (**ε**, **f(ε)**)-approx.

Usually called "Min-2Lin(2)-Deletion".

Let me just call this **2Lin**.

Unratio state of affairs

[easy]: (ε, ε)-approx **NP**-hard

Unratio state of affairs

[easy]: (ε, ε)-approx **NP**-hard

[Håstad]+[TSSW]: (ε, $^5/_4 * \varepsilon$)-approx **NP**-hard

Unratio state of affairs

[easy]: (ε, ε)-approx **NP**-hard

[Håstad]+[TSSW]: (ε, $^5/_4 * \varepsilon$)-approx **NP**-hard

[KKMO]+[MOO]: (ε, $O(\varepsilon^{1/2})$)-approx **UG**-hard

Unratio state of affairs

[easy]: (ε, ε)-approx **NP**-hard

[Håstad]+**[TSSW]**: (ε, $^5/_4 * \varepsilon$)-approx **NP**-hard

[KKMO]+**[MOO]**: (ε, $O(\varepsilon^{1/2})$)-approx **UG**-hard

[GW]: (ε, $O(\varepsilon^{1/2})$)-approx algorithm

Unratio state of affairs

[easy]: (ε, ε)-approx **NP**-hard

[Håstad]+[TSSW]: (ε, $\frac{5}{4}*\varepsilon$)-approx **NP**-hard

[KKMO]+[MOO]: (ε, $O(\varepsilon^{1/2})$)-approx **UG**-hard

[GW]: (ε, $O(\varepsilon^{1/2})$)-approx algorithm

asymptotically off from the truth

Unratio state of affairs

[easy]: (**ε**, **ε**)-approx **NP**-hard

[Håstad]+**[TSSW]**: (**ε**, $^5/_4$***ε**)-approx **NP**-hard

[KKMO]+**[MOO]**: (**ε**, O(**ε**$^{1/2}$))-approx **UG**-hard

[GW]: (**ε**, O(**ε**$^{1/2}$))-approx algorithm

asymptotically off from the truth

[Rao]: If (**ε**, O(f(**q**)***ε**$^{1/2}$))-approx is **NP**-hard for 2Lin(**q**), for f(**q**) = Ω(1), then **UG** is true.

This work

[Håstad]+[TSSW]: (ε, $^5/_4 * \varepsilon$)-approx **NP**-hard

This work

[Håstad]+[TSSW]: (ε, $^5/_4 * \varepsilon$)-approx **NP**-hard

[Us]: (ε, $^{11}/_8 * \varepsilon$)-approx **NP**-hard

This work

[Håstad]+**[TSSW]**: (**ε**, 1.25***ε**)-approx **NP**-hard

[Us]: (**ε**, 1.375***ε**)-approx **NP**-hard

This work

[Håstad]+**[TSSW]**: (ε, $^5/_4 * \varepsilon$)-approx **NP**-hard

[Us]: (ε, $^{11}/_8 * \varepsilon$)-approx **NP**-hard

This work

[Håstad]+**[TSSW]**: (**ε**, $^5/_4$***ε**)-approx **NP**-hard

[Us]: (**ε**, $^{11}/_8$***ε**)-approx **NP**-hard

Cons:

- Still haven't proven **UniqueGames**. 😞

This work

[Håstad]+[TSSW]: $(\varepsilon, \frac{5}{4}*\varepsilon)$-approx **NP**-hard

[Us]: $(\varepsilon, \frac{11}{8}*\varepsilon)$-approx **NP**-hard

Cons:

- Still haven't proven **UniqueGames**. 😞

Pros:

- First improvement since 1997.

This work

[Håstad]+[TSSW]: (ε, $^{5}/_{4}*\varepsilon$)-approx **NP**-hard

[Us]: (ε, $^{11}/_{8}*\varepsilon$)-approx **NP**-hard

Cons:

- Still haven't proven **UniqueGames**. 😞

Pros:

- First improvement since 1997.
- Study new type of "gadget reduction"

This work

[Håstad]+[TSSW]: (ε, $^{5}/_{4}*\varepsilon$)-approx **NP**-hard

[Us]: (ε, $^{11}/_{8}*\varepsilon$)-approx **NP**-hard (**and more!**)

Cons:

- Still haven't proven **UniqueGames**. 😞

Pros:

- First improvement since 1997.
- Study new type of "gadget reduction"

Proving (ε, $\frac{5}{4}$*ε)-hardness

Standard two-step plan.

Proving (ε, $\frac{5}{4} * \varepsilon$)-hardness

Standard two-step plan.

[Håstad]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: $\text{val}(\mathcal{I}) \geq (1 - \varepsilon)$
- **No**: $\text{val}(\mathcal{I}) \leq (\frac{1}{2} + \varepsilon)$

Proving (ε, $^5/_4 * \varepsilon$)-hardness

Standard two-step plan.

[Håstad]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: $\text{val}(\mathcal{I}) \geq (1 - \varepsilon)$
- **No**: $\text{val}(\mathcal{I}) \leq (\frac{1}{2} + \varepsilon)$

(In our language, (ε, ½ - ε)-approxing 3Lin is **NP**-hard.)

Proving (ε, ⁵/₄*ε)-hardness

Standard two-step plan.

[Håstad]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) ≥ (1 - **ε**)
- **No**: val($\mathcal{I}$) ≤ (½ + **ε**)

(In our language, (**ε**, ½ - **ε**)-approxing 3Lin is **NP**-hard.)

Step 2: gadget reduce 3Lin to 2Lin **[TSSW]**

Proving (ε, $^5/_4 * \varepsilon$)-hardness

Standard two-step plan.

[Håstad]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) ≥ (1 - ε)
- **No**: val($\mathcal{I}$) ≤ (½ + ε)

(In our language, (ε, ½ - ε)-approxing 3Lin is **NP**-hard.)

Step 2: gadget reduce 3Lin to 2Lin **[TSSW]**

(see also **[OW12]**)

3Lin

$$x_1 x_3 x_5 = 1$$

$$x_{10} x_{16} x_3 = -1$$

...

$$x_{47} x_{11} x_{98} = -1$$

3Lin

$$x_1 x_3 x_5 = 1$$

$$x_{10} x_{16} x_3 = -1$$

...

$$x_{47} x_{11} x_{98} = -1$$

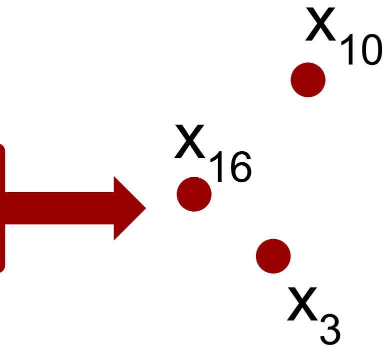
3Lin

$x_1 x_3 x_5 = 1$

$x_{10} x_{16} x_3 = -1$

...

$x_{47} x_{11} x_{98} = -1$



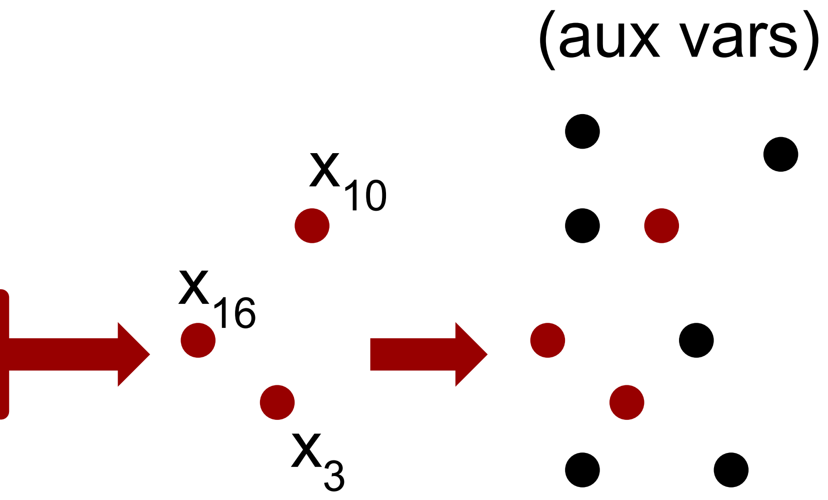
3Lin

$x_1 x_3 x_5 = 1$

$x_{10} x_{16} x_3 = -1$

...

$x_{47} x_{11} x_{98} = -1$



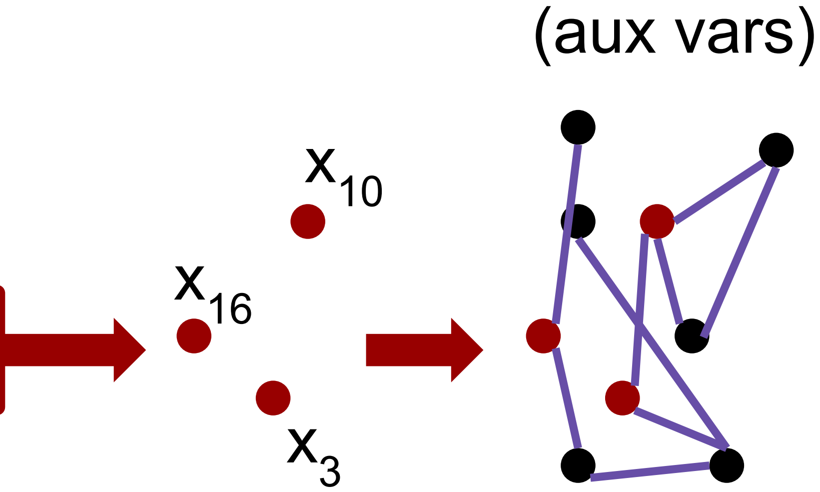
3Lin

$$x_1 x_3 x_5 = 1$$

$$x_{10} x_{16} x_3 = -1$$

...

$$x_{47} x_{11} x_{98} = -1$$



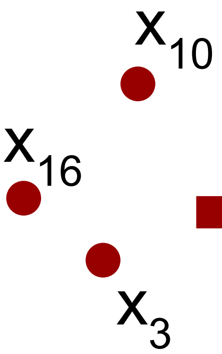
3Lin

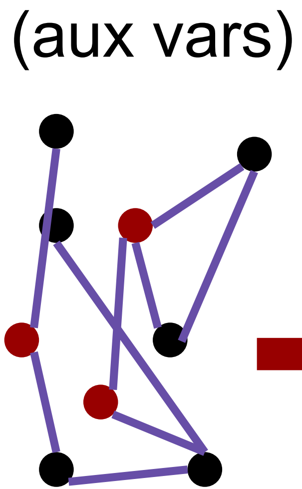
$x_1 x_3 x_5 = 1$

$x_{10} x_{16} x_3 = -1$

...

$x_{47} x_{11} x_{98} = -1$





2Lin gadget

$x_{10} = -x_3$

$y_{61} = -y_{24}$

...

$x_{16} = -y_5$

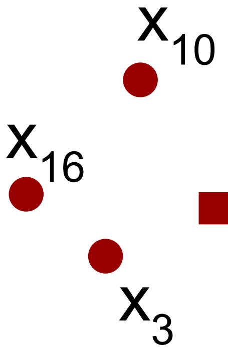
3Lin

$x_1x_3x_5 = 1$

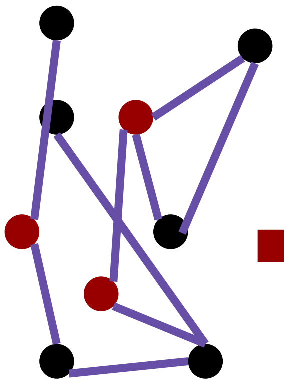
$x_{10}x_{16}x_3 = -1$

…

$x_{47}x_{11}x_{98} = -1$



(aux vars)



2Lin gadget

$x_{10} = -x_3$

$y_{61} = -y_{24}$

…

$x_{16} = -y_5$

Final 2Lin inst: union all the gadgets

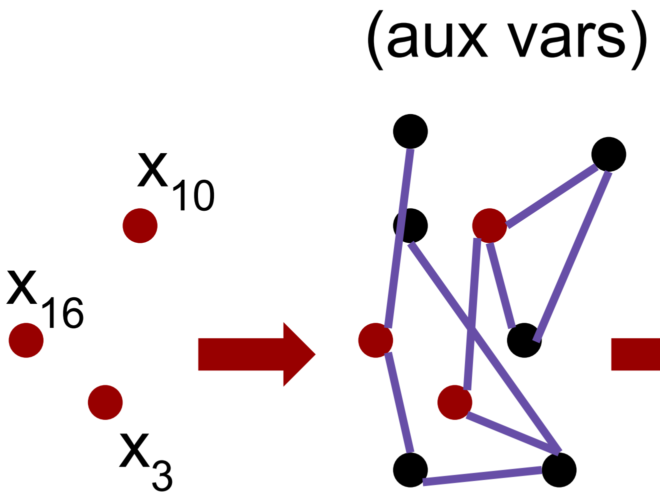
3Lin

$x_1 x_3 x_5 = 1$

$x_{10} x_{16} x_3 = -1$

...

$x_{47} x_{11} x_{98} = -1$



2Lin gadget

$x_{10} = -x_3$

$y_{61} = -y_{24}$

...

$x_{16} = -y_5$

Final 2Lin inst: union all the gadgets

The hope: - x_i's **satisfy** 3Lin eq'n ⇒ good assgn to y_i's

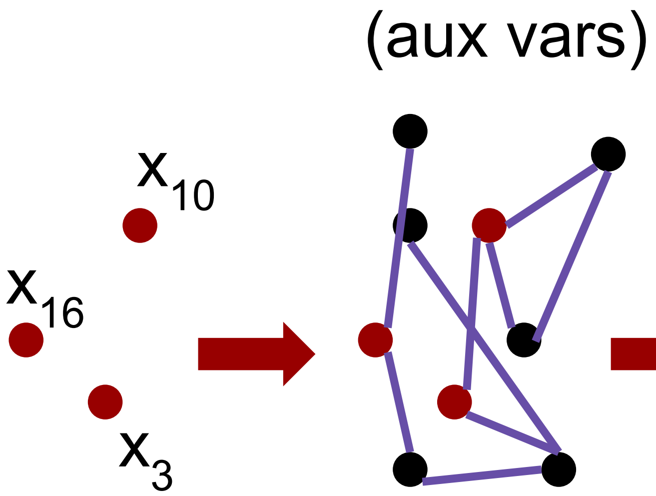
3Lin

$x_1 x_3 x_5 = 1$

$x_{10} x_{16} x_3 = -1$

...

$x_{47} x_{11} x_{98} = -1$



2Lin gadget

$\mathbf{x_{10}} = -\mathbf{x_3}$

$y_{61} = -y_{24}$

...

$\mathbf{x_{16}} = -y_5$

Final 2Lin inst: union all the gadgets

The hope:
- x_i's **satisfy** 3Lin eq'n ⇒ good assgn to y_i's
- x_i's **don't** ⇒ no good assgn to y_i's

Gadgets

Def: A (**c**, **s**)-gadget

Gadgets

Def: A (**c**, **s**)-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)

Gadgets

Def: A (**c**, **s**)-gadget

- **x_i**'s **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- **x_i**'s **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

Gadgets

Def: A (**c**, **s**)-gadget

- **x_i**'s **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- **x_i**'s **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

[TSSW]: there is a 3Lin-to-2Lin (**¼**, **⅜**)-gadget

Gadgets

Def: A (**c**, **s**)-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

[TSSW]: there is a 3Lin-to-2Lin (**¼**, **⅜**)-gadget

(**ε**, ½ - **ε**)-hardess for 3Lin

Gadgets

Def: A (**c**, **s**)-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

[TSSW]: there is a 3Lin-to-2Lin (**¼**, **⅜**)-gadget

(**ε**, ½ - **ε**)-hardess for 3Lin ⇒ - **Yes case**: ¼

Gadgets

Def: A (**c**, **s**)-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

[TSSW]: there is a 3Lin-to-2Lin (**¼**, **⅜**)-gadget

(**ε**, ½ - **ε**)-hardess for 3Lin ⇒
- **Yes case**: ¼
- **No case**: ½ * (¼ + ⅜)

Gadgets

Def: A (**c**, **s**)-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

[TSSW]: there is a 3Lin-to-2Lin (¼, ⅜)-gadget

(**ε**, ½ - **ε**)-hardess for 3Lin ⇒
- **Yes case**: ¼
- **No case**: ½ * (¼ + ⅜) $= {}^{5}/_{16}$

Gadgets

Def: A (**c**, **s**)-gadget

- **x_i**'s **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- **x_i**'s **don't** ⇒ no assgn to y_i's beats value (1 - **s**)

[TSSW]: there is a 3Lin-to-2Lin (**¼**, **⅜**)-gadget

(**ε**, ½ - **ε**)-hardess for 3Lin ⇒

- **Yes case**: ¼
- **No case**: ½ * (¼ + ⅜)

$= {}^{5}/_{16} = {}^{5}/_{4} * \frac{1}{4}$

How do you find gadgets?

Gadgets are just 2Lin instances, so can just monkey around with small instances.

How do you find gadgets?

Gadgets are just 2Lin instances, so can just monkey around with small instances.

More principled: **[TSSW]** show that the optimal gadget can be found via **linear program**!

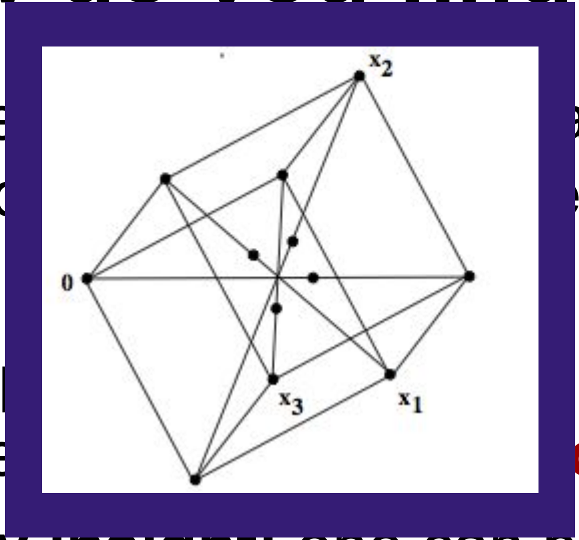
How do you find gadgets?

Gadgets are just 2Lin instances, so can just monkey around with small instances.

More principled: **[TSSW]** show that the optimal gadget can be found via **linear program**!

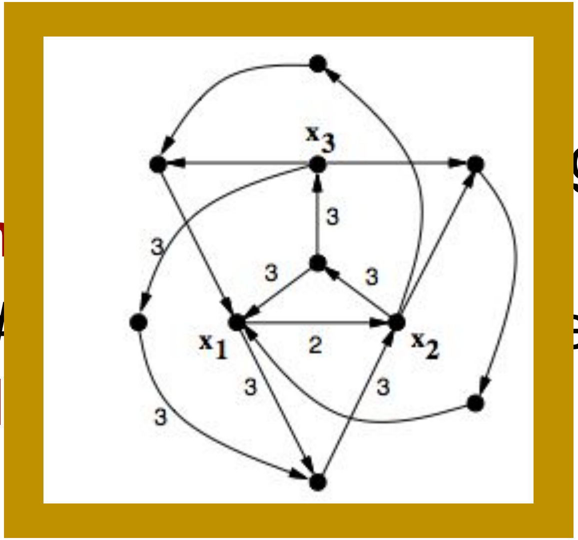
- **key insight:** one can bound # of auxiliary variables
- can certify optimality via **dual LP**.

How do you find gadgets?



Gadge ances, so can just monkey
aroun es.

More show get
can be **ogram**



- **key insight:** one can bound # es
- can certify optimality via **dual**

Our strategy

Old reduction from Håstad's 3Lin hardness result.

Our strategy

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

Our strategy

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

Our strategy

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) = (1 - **ε**)

Our strategy

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) = (1 - **ε**)
- **No**: no matter what assignment, the variables "appear" to be uniformly random

Our strategy

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) = (1 - **ε**)
- **No**: no matter what assignment, the variables "appear" to be uniformly random

Stronger **No** condition. Might help out the gadget.

New gadgets

Def: A (**c**,**s**)-Chan-gadget

New gadgets

Def: A (**c**,**s**)-Chan-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)

New gadgets

Def: A (**c**,**s**)-Chan-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **random** ⇒ on avg., expected best value ≤ (1 - **s**)

New gadgets

Def: A (**c**,**s**)-Chan-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **random** ⇒ on avg., expected best value ≤ (1 - **s**)

Upside: can still solve using an LP

New gadgets

Def: A (**c**,**s**)-Chan-gadget

- x_i's **satisfy** 3Lin eq'n ⇒ an assgn to y_i's of value (1 - **c**)
- x_i's **random** ⇒ on avg., expected best value ≤ (1 - **s**)

Upside: can still solve using an LP

Downside: best 3Lin-to-2Lin gadget no better than in '97!

Our strategy (revised)

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given 3Lin instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) = (1 - **ε**)
- **No**: no matter what assignment, the variables "appear" to be uniformly random

Our strategy (revised)

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given "**balanced pairwise independent subgroup predicate**" instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) = (1 - **ε**)
- **No**: no matter what assignment, the variables "appear" to be uniformly random

Our strategy (revised)

Old reduction from Håstad's 3Lin hardness result.

We now have **better** starting points.

[Chan]: Given "**balanced pairwise independent subgroup predicate**" instance $\mathcal{I}$, **NP**-hard to distinguish

- **Yes**: val($\mathcal{I}$) = (1 - **ε**)
- **No**: no matter what assignment, the variables "appear" to be uniformly random

We instantiate with **BPISP** = $\text{Had}_{\mathbf{k}}$, specifically **k** = 3.

One of Chan's problems

$\textbf{Had}_{\textbf{3}}(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}) = 1$ iff

One of Chan’s problems

$\mathbf{Had_3}(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3})$ = 1 iff

- x_i’s allowed to be arbitrary

One of Chan's problems

Had_3 $(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3})$ = 1 iff

- x_i's allowed to be arbitrary
- $x_{a,b} = x_a \cdot x_b$

One of Chan's problems

$\mathbf{Had_3}$ $(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3})$ = 1 iff

- x_i's allowed to be arbitrary
- $x_{a,b} = x_a \cdot x_b$
- $x_{1,2,3} = x_1 \cdot x_2 \cdot x_3$

One of Chan's problems

$\mathbf{Had_3}(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}) = 1$ iff

- x_i's allowed to be arbitrary
- $x_{a,b} = x_a \cdot x_b$
- $x_{1,2,3} = x_1 \cdot x_2 \cdot x_3 = x_{1,2} \cdot x_3$

One of Chan’s problems

$\mathbf{Had_3}$ $(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3})$ = 1 iff

- x_i’s allowed to be arbitrary
- $x_{a,b} = x_a \cdot x_b$
- $x_{1,2,3} = x_1 \cdot x_2 \cdot x_3 = x_{1,2} \cdot x_3 = x_{1,3} \cdot x_2 = x_1 \cdot x_{2,3}$

One of Chan's problems

$\mathbf{Had_3}$ $(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}) = 1$ iff

- x_i's allowed to be arbitrary
- $x_a \cdot x_b \cdot x_{a,b} = 1$
- $x_{1,2,3} = x_1 \cdot x_2 \cdot x_3 = x_{1,2} \cdot x_3 = x_{1,3} \cdot x_2 = x_1 \cdot x_{2,3}$

One of Chan's problems

$\mathbf{Had_3}$($x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}$) = 1 iff

- x_i's allowed to be arbitrary
- $x_a \cdot x_b \cdot x_{a,b} = 1$
- $x_a \cdot x_{\{1,2,3\}\setminus a} \cdot x_{1,2,3} = 1$ for all $a \in \{1,2,3\}$

One of Chan's problems

Had$_3$($x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}$) = 1 iff

- x_i's allowed to be arbitrary
- $x_a \cdot x_b \cdot x_{a,b} = 1$
- $x_a \cdot x_{\{1,2,3\}\backslash a} \cdot x_{1,2,3} = 1$ for all $a \in \{1,2,3\}$

Contains many simultaneous 3Lin tests. Difficult to satisfy!

One of Chan's problems

$\mathbf{Had_3}(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}) = 1$ iff

- x_i's allowed to be arbitrary
- $x_a \cdot x_b \cdot x_{a,b} = 1$
- $x_a \cdot x_{\{1,2,3\}\setminus a} \cdot x_{1,2,3} = 1$ for all $a \in \{1,2,3\}$

Contains many simultaneous 3Lin tests. Difficult to satisfy!

(not too hard to generalize to $\mathbf{Had_k}$)

One of Chan's problems

$\mathbf{Had_3}(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}) = 1$ iff

- x_i's allowed to be arbitrary
- $x_a \cdot x_b \cdot x_{a,b} = 1$
- $x_a \cdot x_{\{1,2,3\}\setminus a} \cdot x_{1,2,3} = 1$ for all $a \in \{1,2,3\}$

Contains many simultaneous 3Lin tests. Difficult to satisfy!

(not too hard to generalize to $\mathbf{Had_k}$)

[Us]: there is a $(^1/_8, {}^{11}/_{64})$-Chan-gadget from $\mathbf{Had_3}$ to **2Lin**

One of Chan's problems

$\mathbf{Had_3}$ $(x_1, x_2, x_3, x_{1,2}, x_{1,3}, x_{2,3}, x_{1,2,3}) = 1$ iff

- x_i's allowed to be arbitrary
- $x_a \cdot x_b \cdot x_{a,b} = 1$
- $x_a \cdot x_{\{1,2,3\}\backslash a} \cdot x_{1,2,3} = 1$ for all $a \in \{1,2,3\}$

Contains many simultaneous 3Lin tests. Difficult to satisfy!

(not too hard to generalize to **$\mathbf{Had_k}$**)

[Us]: there is a $({}^{1}/_{8}, {}^{11}/_{8} \cdot {}^{1}/_{8})$-Chan-gadget from **$\mathbf{Had_3}$** to **2Lin**

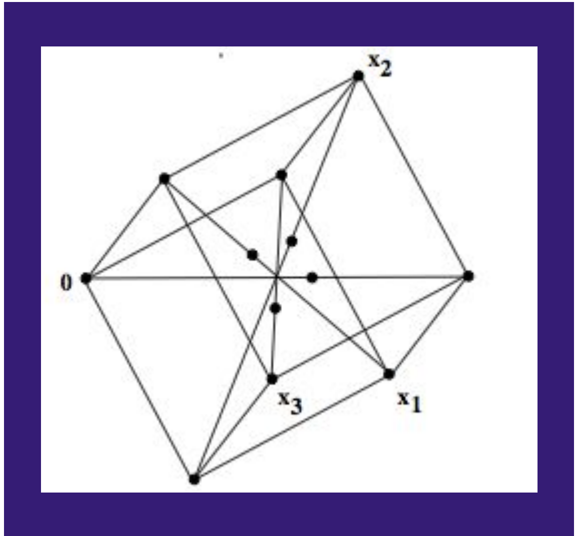
Solving the LP

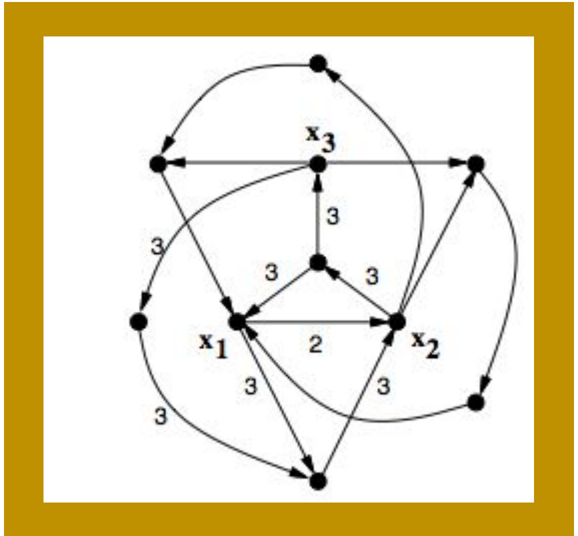
[TSSW]: optimal gadget only needs $2^7 = 128$ variables

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

(so **no** pictures like these)





Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

LP has to consider **all possible** $2^{128} = 3 \times 10^{38}$ assignments to these variables: too large to be feasible!

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

LP has to consider **all possible** $2^{128} = 3 \times 10^{38}$ assignments to these variables: too large to be feasible!

So…….

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

LP has to consider **all possible** $2^{128} = 3 \times 10^{38}$ assignments to these variables: too large to be feasible!

So…….

- lots of work by hand

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

LP has to consider **all possible** $2^{128} = 3 \times 10^{38}$ assignments to these variables: too large to be feasible!

So…….

- lots of work by hand
- lots of computer simulation (/brute force searching)

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

LP has to consider **all possible** $2^{128} = 3 \times 10^{38}$ assignments to these variables: too large to be feasible!

So…….

- lots of work by hand
- lots of computer simulation (/brute force searching)
- lots more work by hand

Solving the LP

[TSSW]: optimal gadget only needs $2^7 = 128$ variables

LP has to consider **all possible** $2^{128} = 3 \times 10^{38}$ assignments to these variables: too large to be feasible!

So…….

- lots of work by hand
- lots of computer simulation (/brute force searching)
- lots more work by hand

but (spoiler alert) it all works out in the end.

Full statement of our results

- (ε, $^{11}/_{8}*\varepsilon$)-approx **NP**-hard (for 2Lin and Max-Cut)

Full statement of our results

- $(\varepsilon, {}^{11}/_{8} * \varepsilon)$-approx **NP**-hard (for 2Lin and Max-Cut)

- $({}^{1}/_{8}, {}^{11}/_{8} \cdot {}^{1}/_{8})$-Chan-gadget from **$\mathbf{Had_3}$** to **2Lin**

Full statement of our results

- (ε, $^{11}/_{8}*\varepsilon$)-approx **NP**-hard (for 2Lin and Max-Cut)

- ($^{1}/_{8}$, $^{11}/_{8}\cdot{}^{1}/_{8}$)-Chan-gadget from **Had$_3$** to **2Lin**
- prove optimality of this gadget via dual solution

Full statement of our results

- (ε, ${}^{11}/_{8}*\varepsilon$)-approx **NP**-hard (for 2Lin and Max-Cut)

- (${}^{1}/_{8}$, ${}^{11}/_{8} \cdot {}^{1}/_{8}$)-Chan-gadget from **Had$_3$** to **2Lin**
- prove optimality of this gadget via dual solution

- can't beat (ε, $2.54 \cdot \varepsilon$)-hardness via a Chan gadget starting from ***any*** **BPISP** predicate

Open problems

We give an optimal gadget reduction from Had_3 to 2Lin.

Open problems

We give an optimal gadget reduction from Had_3 to 2Lin.

We give a gadget from Had_k to 2Lin, along with a **Game Show Conjecture** which would imply (ε, 1.5 · ε)-hardness.

Open problems

We give an optimal gadget reduction from Had_3 to 2Lin.

We give a gadget from Had_k to 2Lin, along with a **Game Show Conjecture** which would imply (ε, $1.5 \cdot \varepsilon$)-hardness.

We couldn't say anything about the normal hardness ratio.

Maybe you can?


Thanks!