Optimal Inapproximability of Max CSPs over large alphabet

Pasin Manurangsi\textsuperscript{1} \hspace{1cm} Preetum Nakkiran\textsuperscript{2} \hspace{1cm} Luca Trevisan\textsuperscript{1}

\textsuperscript{1}UC Berkeley \hspace{1cm} \textsuperscript{2}Harvard

RANDOM-APPROX 2016
Max $k$-CSP$_R$

Maximum Constraint Satisfaction Problem:

- Variables take values in alphabet of size $R$.
- Constraints involve $k$ variables each.
- Goal: find assignment maximizing the number of satisfied constraints.
Max $k$-CSP$_R$

Maximum Constraint Satisfaction Problem:

- Variables take values in alphabet of size $R$.
- Constraints involve $k$ variables each.
- Goal: find assignment maximizing $\#$ of satisfied constraints.

Example

For $k = 2$, $R = 3$, a 2-CSP$_3$ is given by a list of constraints:

\[
\begin{align*}
(x_1 = 0 \land x_2 = 2) \\
(x_1 = 1 \land x_3 = 2) \\
\ldots
\end{align*}
\]
Hardness of $\text{Max } k\text{-CSP}_R$

NP-hard to solve exactly (contains MAX-CUT, MAX 3-SAT).
Hardness of \( \text{Max } k\text{-CSP}_R \)

NP-hard to solve exactly (contains MAX-CUT, MAX 3-SAT).

NP-hard to approximate (PCP theorem).
Hardness of $\text{Max } k\text{-CSP}_R$

NP-hard to solve exactly (contains MAX-CUT, MAX 3-SAT).
np-hard to approximate (PCP theorem).
Boolean CSPs ($R = 2$): Optimal approximation factor is $O(k/2^k)$. 
Hardness of $\text{Max } k\text{-CSP}_R$

NP-hard to solve exactly (contains MAX-CUT, MAX 3-SAT).

NP-hard to approximate (PCP theorem).

Boolean CSPs ($R = 2$): Optimal approximation factor is $O(k/2^k)$.

Non-boolean CSPs ($R > 2$): not resolved prior.
Trivial \((1/R^k)\)-approximation for \(\text{MAX } k\text{-CSP}_R\): Random assignment. Each clause matches the maximizing assignment w.p. \(1/R^k\).

Q: Can we do better? Is it hard to do much better?
Prior Work: Non-boolean Max CSP

Approximation factors:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\frac{\log R}{R}$</td>
<td></td>
</tr>
<tr>
<td>$k = 3$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$3 \leq k < O(1)$

$^1$Ignoring constants, and for large $R$. 
Prior Work: Non-boolean Max CSP

Approximation factors:\(^1\)

<table>
<thead>
<tr>
<th>$k$</th>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{\sqrt{R}}$</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{\sqrt{R}}$</td>
</tr>
</tbody>
</table>

$3 \leq k < O(1)$

\(^1\)Ignoring constants, and for large $R$. 
Prior Work: Non-boolean Max CSP

Approximation factors:\(^1\)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k = 2)</td>
<td>(\frac{\log R}{R})</td>
<td>(\frac{\log R}{R})</td>
</tr>
<tr>
<td>(k = 3)</td>
<td>(\frac{1}{R^2})</td>
<td></td>
</tr>
</tbody>
</table>

\(3 \leq k < O(1)\)

\(^1\)Ignoring constants, and for large \(R\).
Prior Work: Non-boolean Max CSP

Approximation factors:\(^1\)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k = 2)</td>
<td>(\frac{\log R}{R})</td>
<td>(\frac{\log R}{R})</td>
</tr>
<tr>
<td>(k = 3)</td>
<td>(\frac{1}{R^2})</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3 \leq k &lt; O(1))</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\)Ignoring constants, and for large \(R\).
## Prior Work: Non-boolean Max CSP

Approximation factors:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{R}$</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>$\frac{1}{R^2}$</td>
<td>$\frac{1}{R}$</td>
</tr>
<tr>
<td>$3 \leq k &lt; O(1)$</td>
<td>$\frac{1}{R^{k-1}}$</td>
<td>$\frac{1}{R^{k-2}}$</td>
</tr>
</tbody>
</table>

\[^1\text{Ignoring constants, and for large } R.\]
Prior Work: Non-boolean Max CSP

Approximation factors:

<table>
<thead>
<tr>
<th>$k$</th>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\log R / R$</td>
<td>$\log R / R$</td>
<td>$\log R / \sqrt{R}$</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>$1 / R^2$</td>
<td></td>
<td>$1 / R$</td>
</tr>
<tr>
<td>$3 \leq k &lt; O(1)$</td>
<td>$1 / R^{k-1}$</td>
<td></td>
<td>$1 / R^{k-2}$</td>
</tr>
</tbody>
</table>

For constant $k \geq 3$, factor of $R$ gap in hardness vs. approximation.

---

1Ignoring constants, and for large $R$. 
Our results

<table>
<thead>
<tr>
<th>$k$</th>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\log \frac{R}{R}$</td>
<td>$\log \frac{R}{R}$</td>
<td>$\log \frac{R}{\sqrt{R}}$</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>$\frac{1}{R^2}$</td>
<td>$\frac{1}{R}$</td>
<td>$\frac{1}{R}$</td>
</tr>
<tr>
<td>$3 \leq k &lt; O(1)$</td>
<td>$\frac{1}{R^{k-1}}$</td>
<td>$\frac{1}{R^{k-2}}$</td>
<td>$\frac{1}{R^{k-2}}$</td>
</tr>
</tbody>
</table>

The original paper had polylog($R$) gap. Improvement suggested by Rishi Saket, Subhash Khot, Venkat Guriswami.
### Our results

<table>
<thead>
<tr>
<th>$k$</th>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\log \frac{R}{R}$</td>
<td>$\log \frac{R}{R}$</td>
<td>$\log \frac{R}{\sqrt{R}}$</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>$\frac{1}{R^2} \log \frac{R}{R^2}$</td>
<td>$\frac{1}{R^2} \log \frac{R}{R^2}$</td>
<td>$\frac{1}{R}$</td>
</tr>
<tr>
<td>$3 \leq k &lt; O(1)$</td>
<td>$\frac{1}{R^{k-1}} \log \frac{R}{R^{k-1}}$</td>
<td>$\frac{1}{R^{k-1}} \log \frac{R}{R^{k-1}}$</td>
<td>$\frac{1}{R^{k-2}}$</td>
</tr>
</tbody>
</table>

We give matching UG-hardness and approximation algorithms for any $k$, $R$. Gap reduced to $O(1)$ for constant $k$. Original paper had polylog($R$) gap. Improvement suggested by Rishi Saket, Subhash Khot, Venkat Guriswami.
Our results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{R}$</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>$\frac{1}{R^2}$ $\frac{\log R}{R^2}$</td>
<td>$\frac{\log R}{R^2}$</td>
</tr>
<tr>
<td>$3 \leq k &lt; O(1)$</td>
<td>$\frac{1}{R^{k-1}}$ $\frac{\log R}{R^{k-1}}$</td>
<td>$\frac{\log R}{R^{k-1}}$</td>
</tr>
</tbody>
</table>
Our results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>UG-Hardness</th>
<th>NP-Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$</td>
<td>$\frac{\log R}{R}$</td>
<td>$\frac{\log R}{\sqrt{R}}$</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>$\frac{1}{R^2} \frac{\log R}{R^2}$</td>
<td>$\frac{\log R}{R^2}$</td>
</tr>
<tr>
<td>$3 \leq k &lt; O(1)$</td>
<td>$\frac{1}{R^{k-1}} \frac{\log R}{R^{k-1}}$</td>
<td>$\frac{\log R}{R^{k-1}}$</td>
</tr>
</tbody>
</table>

We give matching UG-hardness and approximation algorithms for any $k, R$. Gap reduced to $O(1)$ for constant $k$.\(^2\)

\(^2\) Original paper had $\text{polylog}(R)$ gap. Improvement suggested by Rishi Saket, Subhash Khot, Venkat Guruswami.
Dictator Testing

UG-Hardness-of-approximation equivalent to dictator testing.

Dictator: $f(x_1, x_2, \ldots, x_n) = x_i$. 
Dictator Testing

UG-Hardness-of-approximation equivalent to dictator testing.

Dictator: \( f(x_1, x_2, \ldots, x_n) = x_i \).

Problem

*Given oracle access to \( f : [R]^n \rightarrow [R] \), determine if \( f \) is a dictator or “far from a dictator”.*
Dictator Testing

UG-Hardness-of-approximation equivalent to dictator testing.

Dictator: \( f(x_1, x_2, \ldots, x_n) = x_i \).

Problem

Given oracle access to \( f : [R]^n \rightarrow [R] \), determine if \( f \) is a dictator or “far from a dictator”.

- **Completeness** \( c \): If \( f \) is a dictator, accept w.p. \( \geq c \).
- **Soundness** \( s \): If \( f \) is “far from” a dictator, accept w.p. \( \leq s \).

“Far from dictator” \( \equiv \) small low-degree influences (Fourier condition)
Examples

\[ f : [R]^n \rightarrow R \]

“Far from dictator” ≡ small low-degree influences

Example

Plurality on \( n \) coordinates is far from a dictator (no influential coordinate).
Examples

\[ f : [R]^n \to R \]

“Far from dictator” \(\equiv\) small low-degree influences

Example

Plurality on \(n\) coordinates is far from a dictator (no influential coordinate).

Example

\[ f(x_1, x_2, \ldots, x_n) := x_1 \oplus_R x_2 \]

is NOT far from a dictator.
UG-Hardness of Approximation

\[ k \text{-query dictator test over alphabet } R, \text{ with } (\text{soundness, completeness}) = (s, c) \quad \iff \quad \text{UG-hard to distinguish between } k\text{-CSP}_R \text{ instances where } \OPT \approx s \text{ vs. } \OPT \approx c \]
UG-Hardness of Approximation

\[ k \text{-query dictator test over alphabet } R, \text{ with } (\text{soundness, completeness}) = (s, c) \quad \iff \quad \text{UG-hard to distinguish between } k\text{-CSP}_R \text{ instances where } OPT \approx s \text{ vs. } OPT \approx c \]

\[ \Downarrow \]

\text{UG-hard to approximate Max } k\text{-CSP}_R \text{ better than } \approx (s/c). \]
UG-hardness of boolean 2-CSP

[Khot, Kindler, Mossel, O'Donnell]

2-Query Boolean Dictator test

\( f : \{0, 1\}^n \to \{0, 1\}, \quad \mathbb{E}[f] = 1/2. \)
UG-hardness of boolean 2-CSP

[Khot, Kindler, Mossel, O'Donnell]

2-Query Boolean Dictator test

\[ f : \{0, 1\}^n \rightarrow \{0, 1\}, \quad E[f] = 1/2. \]

- Pick \( x \sim \{0, 1\}^n \) uniform
UG-hardness of boolean 2-CSP

[Khot, Kindler, Mossel, O'Donnell]

2-Query Boolean Dictator test

\[ f : \{0, 1\}^n \to \{0, 1\}, \quad \mathbb{E}[f] = 1/2. \]

- Pick \( x \sim \{0, 1\}^n \) uniform
- Pick “noise” \( \eta \sim \{0, 1\}^n \), each coordinate \( \text{Bernoulli}(p) \).
UG-hardness of boolean 2-CSP
[Khot, Kindler, Mossel, O’Donnell]

2-Query Boolean Dictator test

\( f : \{0, 1\}^n \rightarrow \{0, 1\}, \quad \mathbb{E}[f] = 1/2. \)

- Pick \( x \sim \{0, 1\}^n \) uniform
- Pick “noise” \( \eta \sim \{0, 1\}^n \), each coordinate \( \text{Bernoulli}(p) \).
- Accept iff \( f(x) = f(x \oplus \eta) \)
UG-hardness of boolean 2-CSP
[Khot, Kindler, Mossel, O’Donnell]

2-Query Boolean Dictator test

\( f : \{0, 1\}^n \rightarrow \{0, 1\}, \quad \mathbb{E}[f] = 1/2. \)

- Pick \( x \sim \{0, 1\}^n \) uniform
- Pick “noise” \( \eta \sim \{0, 1\}^n \), each coordinate \( \text{Bernoulli}(p) \).
- Accept iff \( f(x) = f(x \oplus \eta) \)

For \( p \approx 0.15 \),

- **Completeness:** If \( f \) is a dictator, accepts w.p. \( \geq 1 - p \approx 0.85. \)
- **Soundness:** If \( f \) is “far from” a dictator, accepts w.p. \( \leq \approx 0.74. \)
UG-hardness of boolean 2-CSP

[Khot, Kindler, Mossel, O’Donnell]

2-Query Boolean Dictator test

\[ f : \{0, 1\}^n \to \{0, 1\}, \quad \mathbb{E}[f] = 1/2. \]

- Pick \( x \sim \{0, 1\}^n \) uniform
- Pick “noise” \( \eta \sim \{0, 1\}^n \), each coordinate \( \text{Bernoulli}(p) \).
- Accept iff \( f(x) = f(x \oplus \eta) \)

For \( p \approx 0.15 \),

- **Completeness:** If \( f \) is a dictator, accepts w.p. \( \geq 1 - p \approx 0.85 \).
- **Soundness:** If \( f \) is “far from” a dictator, accepts w.p. \( \leq \approx 0.74 \).
- **Ratio:** \( s/c \approx 0.878567 = \alpha_{GW} \)
Why it works

Verifier accepts iff

\[ f(x) = f(x + \eta) \]

Noise \( \eta \) iid on every coordinate.
Why it works

Verifier accepts iff

\[ f(x) = f(x + \eta) \]

Noise \( \eta \) iid on every coordinate.

If \( f \) depends on many coordinates, the noise will “add up”: \( f(x + \eta) \) will be almost uncorrelated with \( f(x) \).
Why it works

Verifier accepts iff

\[ f(x) = f(x + \eta) \]

Noise \( \eta \) iid on every coordinate.

If \( f \) depends on many coordinates, the noise will “add up”: \( f(x + \eta) \) will be almost uncorrelated with \( f(x) \).

Example

majority function \( maj : \{\pm 1\}^n \rightarrow \{\pm 1\} \).

\[ maj(x_1, \ldots, x_n) = sign(\sum_i x_i) \]

If noise \( \eta \) is high enough, \( sign(\sum_i x_i) \) will be almost independent of \( sign(\sum_i (x_i + \eta_i)) \)
Our $k$-query large alphabet dictator test

\[ f : [R]^n \rightarrow [R] \]

$f$ is balanced: All pre-images $f^{-1}(i)$ of same size.
Our $k$-query large alphabet dictator test

$f : [R]^n \to [R]$

$f$ is balanced: All pre-images $f^{-1}(i)$ of same size.

- Pick $z \sim [R]^n$ uniform
Our $k$-query large alphabet dictator test

\[ f : [R]^n \to [R] \]

$f$ is balanced: All pre-images $f^{-1}(i)$ of same size.

- Pick $z \sim [R]^n$ uniform
- Pick $k$ iid noise $\eta_1, \ldots, \eta_k \in [R]^n$, s.t. each coordinate of $\eta_j$ is

\[
\begin{cases} 
0 & \text{w.p. } \rho \\
\text{uniform in } [R] & \text{otherwise}
\end{cases}
\]
Our \( k \)-query large alphabet dictator test

\[ f : [R]^n \rightarrow [R] \]

\( f \) is balanced: All pre-images \( f^{-1}(i) \) of same size.

- Pick \( z \sim [R]^n \) uniform
- Pick \( k \) iid noise \( \eta_1, \ldots, \eta_k \in [R]^n \), s.t. each coordinate of \( \eta_j \) is

\[
\begin{cases} 
0 & \text{w.p. } \rho \\
\text{uniform in } [R] & \text{otherwise}
\end{cases}
\]

- Accept iff \( f(z + \eta_1) = f(z + \eta_2) = \cdots = f(z + \eta_k) \)
Our Results

\[ f : [R]^n \rightarrow [R] \]

Accept iff \( f(z + \eta_1) = f(z + \eta_2) = \cdots = f(z + \eta_k) \)
Our Results

\[ f : [R]^n \to [R] \]

Accept iff \( f(z + \eta_1) = f(z + \eta_2) = \cdots = f(z + \eta_k) \)

We show:

- **Completeness:** If \( f \) is a dictator (\( f(x) = x_j \)), accepts w.p. \( \approx \frac{1}{(\log R)^{k/2}} \)

- **Soundness:** If \( f \) is balanced and has small influences, accepts w.p. \( \leq \approx \frac{1}{R^{k-1}} \)

If \( f \) is far from dictator, the \( k \) queries \( f(z + \eta_1), f(z + \eta_2), \ldots \) look almost independent – all equal w.p. \( \approx \frac{1}{R^{k-1}} \).
Soundness Analysis Ideas

Define \( f^i : [R]^n \rightarrow \{0, 1\} \) as \( f^i(x) := \mathbb{1}[f(x) = i] \)
Soundness Analysis Ideas

Define \( f^i : [R]^n \rightarrow \{0, 1\} \) as \( f^i(x) := 1[f(x) = i] \)

\( \mathbb{E}[f^i] = 1/R \) since \( f \) is balanced.
Soundness Analysis Ideas

Define $f^i : [R]^n \rightarrow \{0, 1\}$ as $f^i(x) := 1[f(x) = i]$

$\mathbb{E}[f^i] = 1/R$ since $f$ is balanced.

$$\Pr[\text{accept}] = \Pr[f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$$
Soundness Analysis Ideas

Define \( f^i : [R]^n \rightarrow \{0, 1\} \) as \( f^i(x) := 1[f(x) = i] \)

\( \mathbb{E}[f^i] = 1/R \) since \( f \) is balanced.

\[
\Pr[\text{accept}] = \Pr[f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]
\]
\[
= \sum_{i \in [R]} \Pr[i = f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]
\]
Soundness Analysis Ideas

Define $f^i : [R]^n \rightarrow \{0, 1\}$ as $f^i(x) := 1[f(x) = i]$

$\mathbb{E}[f^i] = 1/R$ since $f$ is balanced.

$\Pr[\text{accept}] = \Pr[f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$

$= \sum_{i \in [R]} \Pr[i = f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$

$= \sum_{i \in [R]} \mathbb{E}_{x, \eta}[f^i(x + \eta_1) f^i(x + \eta_2) \cdots f^i(x + \eta_k)]$
Soundness Analysis Ideas

Define $f^i : [R]^n \rightarrow \{0, 1\}$ as $f^i(x) := \mathbb{1}[f(x) = i]$

$\mathbb{E}[f^i] = 1/R$ since $f$ is balanced.

$$Pr[\text{accept}] = Pr[f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$$
$$= \sum_{i \in [R]} Pr[i = f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$$
$$= \sum_{i \in [R]} \mathbb{E}_{x,\eta}[f^i(x + \eta_1)f^i(x + \eta_2)\cdots f^i(x + \eta_k)]$$

Define $g^i(x) = \mathbb{E}_\eta[f^i(x + \eta)]$ (i.e. $g^i := T_\rho f^i$)
Soundness Analysis Ideas

Define $f^i : [R]^n \rightarrow \{0, 1\}$ as $f^i(x) := \mathbb{1}[f(x) = i]$

$\mathbb{E}[f^i] = 1/R$ since $f$ is balanced.

$$\Pr[\text{accept}] = \Pr[f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$$

$$= \sum_{i \in [R]} \Pr[i = f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]$$

$$= \sum_{i \in [R]} \mathbb{E}_{x, \eta}[f^i(x + \eta_1)f^i(x + \eta_2)\cdots f^i(x + \eta_k)]$$

$$= \sum_{i \in [R]} \mathbb{E}_{x}[(g^i(x))^k]$$

Define $g^i(x) = \mathbb{E}_{\eta}[f^i(x + \eta)]$ (i.e. $g^i := T_{\rho}f^i$)
Soundness Analysis Ideas

Define \( f^i : [R]^n \rightarrow \{0, 1\} \) as \( f^i(x) := \mathbb{1}[f(x) = i] \)

\[ \mathbb{E}[f^i] = 1/R \text{ since } f \text{ is balanced.} \]

\[
\Pr[\text{accept}] = \Pr[f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]
= \sum_{i \in [R]} \Pr[i = f(x + \eta_1) = f(x + \eta_2) = \cdots = f(x + \eta_k)]
= \sum_{i \in [R]} \mathbb{E}_{x, \eta}[f^i(x + \eta_1)f^i(x + \eta_2)\cdots f^i(x + \eta_k)]
= \sum_{i \in [R]} \mathbb{E}_x[(g^i(x))^k]

(want) \approx \sum_{i \in [R]} \mathbb{E}_x[(g^i(x))]^k = \sum_i (1/R)^k = 1/R^{k-1}

Define \( g^i(x) = \mathbb{E}_\eta[f^i(x + \eta)] \) (i.e. \( g^i := T_\rho f^i \))
Soundness Analysis Ideas

\[ f : [R]^n \to [0, 1], \quad \mathbb{E}[f] = 1/R \]
Soundness Analysis Ideas

\[ f : \mathbb{R}^n \to [0, 1], \quad \mathbb{E}[f] = 1/R \]

\[ g(x) = \mathbb{E}_\eta[f(x + \eta)] = (T_\rho f)(x) \]
Soundness Analysis Ideas

\[ f : [R]^n \to [0, 1], \quad \mathbb{E}[f] = 1/R \]

\[ g(x) = \mathbb{E}_\eta[f(x + \eta)] = (T_\rho f)(x) \]

Want to show:

\[ \mathbb{E}[(T_\rho f)^k] \preceq \mathbb{E}[f]^k \iff \|T_\rho f\|_k \preceq \|f\|_1 \]
Soundness Analysis Ideas

\[ f : [R]^n \to [0, 1], \quad \mathbb{E}[f] = 1/R \]

\[ g(x) = \mathbb{E}_{\eta}[f(x + \eta)] = (T_\rho f)(x) \]

Want to show:

\[ \mathbb{E}[(T_\rho f)^k] \lesssim \mathbb{E}[f]^k \iff ||T_\rho f||_k \lesssim ||f||_1 \]

This is hypercontractivity
Soundness Analysis Ideas

\[ f : [R]^n \to [0,1], \quad \mathbb{E}[f] = 1/R \]

\[ g(x) = \mathbb{E}_\eta[f(x + \eta)] = (T_\rho f)(x) \]

Want to show:

\[ \mathbb{E}[(T_\rho f)^k] \lesssim \mathbb{E}[f]^k \iff \| T_\rho f \|^k \lesssim \| f \|_1 \]

This is hypercontractivity
Algorithm Ideas

$k$-CSP$_R := k$ variables per clause; variables over domain $[R]$
Algorithm Ideas

\( k \)-CSP\(_R \) := \( k \) variables per clause; variables over domain \([R]\)

\( k = 2 \):

Existing \((\frac{\log R}{R})\)-approximation algo (SDP-based).

Advantage of \( A = R \log R \) over random assignment \((\frac{1}{R^2})\).
Algorithm Ideas

\[ k\text{-CSP}_R := k \text{ variables per clause; variables over domain } [R] \]

\[ k = 2 : \]

Existing \((\frac{\log R}{R})\)-approximation algo (SDP-based).

Advantage of \( A = R \log R \) over random assignment \((\frac{1}{R^2})\).

\[ k \geq 3 : \]

Reduction to \( k = 2 \) case, preserving the Advantage over random assignment.

\[ \implies (\frac{\log R}{R^{k-1}})\text{-approximation algo for } k \geq 3. \]
Algorithm Ideas

$k$-CSP$_R := k$ variables per clause; variables over domain $[R]$

$k = 2$:

Existing $(\log R R)\text{-approximation algo (SDP-based)}$.

Advantage of $A = R \log R$ over random assignment $(\frac{1}{R^2})$.

$k \geq 3$:

Reduction to $k = 2$ case, preserving the Advantage over random assignment.

$\implies \left(\frac{\log R}{R^{k-1}}\right)\text{-approximation algo for } k \geq 3$.

Map a constraint

$$(X_1 = a_1) \land (X_2 = a_2) \land \cdots \land (X_k = a_k)$$

to all pairwise constraints $\{(X_i = a_i \land X_j = a_j) : 1 \leq i < j \leq k\}$
Conclusions

We know the approximability of $\text{Max } k\text{-CSP}_R$ for constant $k$, assuming Unique Games Conjecture: $\Theta\left(\frac{\log R}{R^{k-1}}\right)$.
We know the approximability of $\text{Max } k$-CSP$_R$ for constant $k$, assuming Unique Games Conjecture: $\Theta\left(\frac{\log R}{R^{k-1}}\right)$

Previously we knew it was between $\frac{1}{R^{k-2}}$ and $\frac{1}{R^{k-1}}$ (NP-hardness).
Conclusions

We know the approximability of Max $k$-CSP$_R$ for constant $k$, assuming Unique Games Conjecture: $\Theta(\frac{\log R}{R^{k-1}})$

Previously we knew it was between $\frac{1}{R^{k-2}}$ and $\frac{1}{R^{k-1}}$ (NP-hardness).

Open:

- Tight results for larger $k < R$. (already known for $k \geq R$).
  Current gap is $\frac{\log R}{R^{k-1}}$ vs. $\frac{k^2 \log(kR)}{R^{k-1}}$
Conclusions

We know the approximability of $\text{Max } k$-$\text{CSP}_R$ for constant $k$, assuming Unique Games Conjecture: $\Theta(\frac{\log R}{R^{k-1}})$

Previously we knew it was between $\frac{1}{R^{k-2}}$ and $\frac{1}{R^{k-1}}$ (NP-hardness).

Open:

- Tight results for larger $k < R$. (already known for $k \geq R$).
  Current gap is $\frac{\log R}{R^{k-1}}$ vs. $\frac{k^2 \log(kR)}{R^{k-1}}$
- Tight results based on NP-hardness