Recent Hardness of Approximation Results in Parameterized Complexity

Karthik C. S.

Rutgers University

December 10, 2021

Outline

Part 1: Hardness of Approximation meets Parameterized Complexity

- k-Set Cover
- k-Set Intersection
- k-Clique

Outline

Part 1: Hardness of Approximation meets Parameterized Complexity

- k-Set Cover
- k-Set Intersection
- k-Clique

Part 2: Hardness of Approximating k-Clique

- NP World
- Lin's Insight
- Lin's Result: Constant Inapproximability
- New Result: Almost Polynomial factor Inapproximability

Outline

Part 1: Hardness of Approximation meets Parameterized Complexity

- k-Set Cover
- k-Set Intersection
- k-Clique

Part 2: Hardness of Approximating k-Clique

- NP World
- Lin's Insight
- Lin's Result: Constant Inapproximability
- New Result: Almost Polynomial factor Inapproximability

Part 3: High Level Remarks

Outline

Part 1: Hardness of Approximation meets Parameterized Complexity

- k-Set Cover
- k-Set Intersection
- k-Clique

Part 2: Hardness of Approximating k-Clique

- NP World
- Lin's Insight
- Lin's Result: Constant Inapproximability
- New Result: Almost Polynomial factor Inapproximability

Part 3: High Level Remarks

Part 1

Hardness of Approximation
meets
Parameterized Complexity

Parameterized Inapproximability: Motivation

- Many Optimization problems are NP-Hard

Parameterized Inapproximability: Motivation

- Many Optimization problems are NP-Hard
- Coping mechanisms
  - Approximation Algorithms
  - Fixed Parameter Tractability

Parameterized Inapproximability: Motivation

- Many Optimization problems are NP-Hard
- Coping mechanisms
  - Approximation Algorithms
  - Fixed Parameter Tractability
- Set Cover, Clique, Set Intersection: Hard to cope!

Parameterized Inapproximability: Motivation

- Many Optimization problems are NP-Hard
- Coping mechanisms
  - Approximation Algorithms
  - Fixed Parameter Tractability
- Set Cover, Clique, Set Intersection: Hard to cope!
- New direction: Fixed Parameter Approximability

Is there a $F(k) \cdot \text{poly}(n)$ time algorithm
that approximates to a factor $T(k)$?

k-Set Cover

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose union is $[n]$

k-Set Cover

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose union is $[n]$

- Greedy Algorithm: $\ln n$ approximation (Slavík'96)

k-Set Cover

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose union is $[n]$

- Greedy Algorithm: $\ln n$ approximation (Slavík'96)
- NP-Hard to do better (Dinur-Steurer'14)

k-Set Cover

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose union is $[n]$

- Greedy Algorithm: $\ln n$ approximation (Slavík'96)
- NP-Hard to do better (Dinur-Steurer'14)
- W[2]-complete (Downey-Fellows'95)

k-Set Cover

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose union is $[n]$

- Greedy Algorithm: $\ln n$ approximation (Slavík'96)
- NP-Hard to do better (Dinur-Steurer'14)
- W[2]-complete (Downey-Fellows'95)
- W[1]-hard to approximate to $F(k)$ factor for any F (**K**-Laekhanukit-Manurangsi'18)
- W[1]-hard to approximate to $(\log n)^{1/\varepsilon(k)}$ factor for any unbounded ε (Lin'19)

k-Set Intersection

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose intersection is maximized

k-Set Intersection

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose intersection is maximized

- NP World: Ruling out PTAS (assuming NP$\neq$P) is open!

k-Set Intersection

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose intersection is maximized

- NP World: Ruling out PTAS (assuming NP$\neq$P) is open!
- No poly factor approximation poly time algorithm assuming "weak-ETH" (Xavier'12)
  - Relies on Quasi-random PCP of (Khot'06)

k-Set Intersection

Input: $S_1, \ldots, S_n \subseteq [n]$

Output: $S_{i_1}, \ldots, S_{i_k}$ whose intersection is maximized

- NP World: Ruling out PTAS (assuming NP≠P) is open!
- No poly factor approximation poly time algorithm assuming "weak-ETH" (Xavier'12)
  - Relies on Quasi-random PCP of (Khot'06)
- W[1]-hard to approximate to $n^{o\left(\frac{1}{\sqrt{k}}\right)}$ factor (Lin'15)

k-Clique

Input: $G(V, E)$

Output: k-clique in G

k-Clique

Input: $G(V, E)$

Output: k-clique in G

- Trivial Algorithm: k approximation

k-Clique

Input: $G(V, E)$

Output: k-clique in G

- Trivial Algorithm: k approximation
- NP-Hard to approximate to $n^{1-\varepsilon}$ factor for any $\varepsilon > 0$ (Håstad'96)

k-Clique

Input: $G(V, E)$
Output: k-clique in G

- Trivial Algorithm: k approximation
- NP-Hard to approximate to $n^{1-\varepsilon}$ factor for any $\varepsilon > 0$ (Håstad'96)
- W[1]-complete (Downey-Fellows'95)

k-Clique

Input: $G(V, E)$

Output: k-clique in G

- Trivial Algorithm: k approximation
- NP-Hard to approximate to $n^{1-\varepsilon}$ factor for any $\varepsilon > 0$ (Håstad'96)
- W[1]-complete (Downey-Fellows'95)
- W[1]-complete to approximate to constant factor (Lin'21)

k-Clique

Input: $G(V, E)$

Output: k-clique in G

- Trivial Algorithm: k approximation
- NP-Hard to approximate to $n^{1-\varepsilon}$ factor for any $\varepsilon > 0$ (Håstad'96)
- W[1]-complete (Downey-Fellows'95)
- W[1]-complete to approximate to constant factor (Lin'21)
- W[1]-complete to approximate to $k^{o(1)}$ factor (**K**-Khot'21)

The Big Prize

2-CSP:

The Big Prize

2-CSP:

Input: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

The Big Prize

2-CSP:

Input: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u), \sigma(v)) \in \pi_{(u,v)}]$

The Big Prize

2-CSP:

Input: Graph $G(V,E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u),\sigma(v))\in\pi_{(u,v)}]$

- NP-hard to approximate to 0.99-factor even when $|\Sigma| = 3$
  - PCP Theorem for NP (Arora-Safra'92; Arora-Lund-Motwani-Sudan-Szegedy'92)

The Big Prize

2-CSP:

Input: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u), \sigma(v)) \in \pi_{(u,v)}]$

- NP-hard to approximate to 0.99-factor even when $|\Sigma| = 3$
  - PCP Theorem for NP (Arora-Safra'92; Arora-Lund-Motwani-Sudan-Szegedy'92)
- Parameterized by $|V|$: W[1]-complete

The Big Prize

2-CSP:

Input: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u), \sigma(v)) \in \pi_{(u,v)}]$

- NP-hard to approximate to 0.99-factor even when $|\Sigma| = 3$
  - PCP Theorem for NP (Arora-Safra'92; Arora-Lund-Motwani-Sudan-Szegedy'92)
- Parameterized by $|V|$: W[1]-complete
- Parameterized by $|V|$: W[1]-hard to approximate?

The Big Prize

2-CSP:

Input: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u), \sigma(v)) \in \pi_{(u,v)}]$

- NP-hard to approximate to 0.99-factor even when $|\Sigma| = 3$
  - PCP Theorem for NP (Arora-Safra'92; Arora-Lund-Motwani-Sudan-Szegedy'92)
- Parameterized by $|V|$: W[1]-complete
- Parameterized by $|V|$: W[1]-hard to approximate?
  - PCP Theorem for Parameterized Complexity

The Big Prize

2-CSP:

Input: Graph $G(V,E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u),\sigma(v))\in\pi_{(u,v)}]$

- NP-hard to approximate to 0.99-factor even when $|\Sigma| = 3$
  - PCP Theorem for NP (Arora-Safra'92; Arora-Lund-Motwani-Sudan-Szegedy'92)
- Parameterized by $|V|$: W[1]-complete
- Parameterized by $|V|$: W[1]-hard to approximate?
  - PCP Theorem for Parameterized Complexity

Parameterized Inapproximability Hypothesis (PIH)

(Lokshtanov-Ramanujan-Saurabh-Zehavi'17)

There exists $\delta > 0$ such that 2-CSP on k vertices and alphabet size n is W[1]-hard to approximate to $(1-\delta)$ factor

k-Clique

Theorem (Lin'21)

Approximating k-Clique to any $O(1)$ factor is W[1]-hard.

Theorem (K-Khot'21)

Approximating k-Clique to any $k^{o(1)}$ factor is W[1]-hard.

Part 2

Hardness of Approximating
k-Clique

FGLSS reduction

- FGLSS reduction: first NP-hardness of approximation

  (Feige-Goldwasser-Lovaász-Safra-Szegedy'91)

FGLSS reduction

- FGLSS reduction: first NP-hardness of approximation
  (Feige-Goldwasser-Lovaász-Safra-Szegedy'91)
- Combined with sophisticated PCPs and graph products yields NP-hardness of approximating Clique to $n^{1-\varepsilon}$ factor
  (Samorodnitsky-Trevisan'00)

FGLSS reduction

- FGLSS reduction: first NP-hardness of approximation
  (Feige-Goldwasser-Lovaász-Safra-Szegedy'91)
- Combined with sophisticated PCPs and graph products yields NP-hardness of approximating Clique to $n^{1-\varepsilon}$ factor
  (Samorodnitsky-Trevisan'00)

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

FGLSS reduction

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

FGLSS reduction

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

2-CSP: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

FGLSS reduction

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

2-CSP: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

k-Clique: Graph $H(\{(e, \sigma_e) \mid e \in E, \sigma_e : e \to \Sigma, \sigma_e \in \pi_e\}, F)$, $k := |E|$

FGLSS reduction

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

2-CSP: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

k-Clique: Graph $H(\{(e, \sigma_e) \mid e \in E, \sigma_e : e \to \Sigma, \sigma_e \in \pi_e\}, F)$, $k := |E|$

Edges in H: $(e, \sigma_e : e \to \Sigma)$ and $(e', \sigma'_{e'} : e' \to \Sigma)$ is NOT an edge in H iff $\exists v \in e \cap e'$ such that $\sigma_e(v) \neq \sigma'_{e'}(v)$

FGLSS reduction

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

2-CSP: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

k-Clique: Graph $H(\{(e, \sigma_e) \mid e \in E, \sigma_e : e \to \Sigma, \sigma_e \in \pi_e\}, F)$, $k := |E|$

Edges in H: $(e, \sigma_e : e \to \Sigma)$ and $(e', \sigma'_{e'} : e' \to \Sigma)$ is NOT an edge in H iff $\exists v \in e \cap e'$ such that $\sigma_e(v) \neq \sigma'_{e'}(v)$

Completeness: $\sigma : V \to \Sigma$ satifies all constraints then there is $|E|$ sized clique in H: $\{((u, v), (\sigma(u), \sigma(v))) \mid (u, v) \in E\}$

FGLSS reduction

Theorem

Assuming PIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

2-CSP: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

k-Clique: Graph $H(\{(e, \sigma_e) \mid e \in E, \sigma_e : e \to \Sigma, \sigma_e \in \pi_e\}, F)$, $k := |E|$

Edges in H: $(e, \sigma_e : e \to \Sigma)$ and $(e', \sigma'_{e'} : e' \to \Sigma)$ is NOT an edge in H iff $\exists v \in e \cap e'$ such that $\sigma_e(v) \neq \sigma'_{e'}(v)$

Completeness: $\sigma : V \to \Sigma$ satifies all constraints then there is $|E|$ sized clique in H: $\{((u, v), (\sigma(u), \sigma(v))) \mid (u, v) \in E\}$

Soundness: If $T \subseteq \{(e, \sigma_e) \mid e \in E\}$ is a clique in H of size $(1 - \varepsilon)k$ then we can recover an almost good global assignment

Lin's Observation

Boolean Hypercube Graph $Q_t(V, E)$:

Lin's Observation

Boolean Hypercube Graph $Q_t(V, E)$:

Vertices: $V = \{0, 1\}^t$

Lin's Observation

Boolean Hypercube Graph $Q_t(V, E)$:

Vertices: $V = \{0,1\}^t$

Edges: $(u, v) \in E$ iff u and v differ on 1 coordinate

Lin's Observation

Boolean Hypercube Graph $Q_t(V, E)$:

Vertices: $V = \{0,1\}^t$

Edges: $(u, v) \in E$ iff u and v differ on 1 coordinate

Weak Parameterized Inapproximability Hypothesis (WPIH)

There exists $\delta > 0$ such that given an instance φ of 2-CSP on Q_t over alphabet size n, it is W[1]-hard to distinguish:

Completeness: φ has a satisfying assignment

Soundness: For every assignment σ to φ there exists $i \in [t]$ such that δ fraction of edges in direction i are violated by σ

Lin's Observation

Boolean Hypercube Graph $Q_t(V, E)$:

Vertices: $V = \{0, 1\}^t$

Edges: $(u, v) \in E$ iff u and v differ on 1 coordinate

Weak Parameterized Inapproximability Hypothesis (WPIH)

There exists $\delta > 0$ such that given an instance φ of 2-CSP on Q_t over alphabet size n, it is W[1]-hard to distinguish:

Completeness: φ has a satisfying assignment

Soundness: For every assignment σ to φ there exists $i \in [t]$ such that δ fraction of edges in direction i are violated by σ

Theorem

Assuming WPIH, approximating k-Clique to any $O(1)$ factor is W[1]-hard.

Lin's Proof Outline

Theorem (Lin'21)

Approximating k-Clique to any $O(1)$ factor is W[1]-hard.

Lin's Proof Outline

Theorem (Lin'21)

Approximating k-Clique to any $O(1)$ factor is W[1]-hard.

- Start from k-Vector Sum problem

Lin's Proof Outline

Theorem (Lin'21)

Approximating k-Clique to any $O(1)$ factor is W[1]-hard.

- Start from k-Vector Sum problem
- Prove a weaker version of WPIH

Lin's Proof Outline

Theorem (Lin'21)

Approximating k-Clique to any $O(1)$ factor is W[1]-hard.

- Start from k-Vector Sum problem
- Prove a weaker version of WPIH
- Develop a novel modification of FGLSS reduction

Lin's Proof Outline

k-Vector Sum Problem:

Lin's Proof Outline

k-Vector Sum Problem:

Input: $U_1, \ldots, U_k \subseteq \mathbb{F}_2^{h \log n}, \forall i \in [k], \; |U_i| = n$

Lin's Proof Outline

k-Vector Sum Problem:

Input: $U_1, \ldots, U_k \subseteq \mathbb{F}_2^{h \log n}, \forall i \in [k], \; |U_i| = n$

Output: If there exists $(u_1, \ldots, u_k) \in U_1 \times \cdots \times U_k$ such that $\sum_{i \in [k]} u_i = \vec{0}$

Lin's Proof Outline

k-Vector Sum Problem:

Input: $U_1, \ldots, U_k \subseteq \mathbb{F}_2^{h \log n}, \forall i \in [k], \ |U_i| = n$

Output: If there exists $(u_1, \ldots, u_k) \in U_1 \times \cdots \times U_k$ such that $\sum_{i \in [k]} u_i = \vec{0}$

Theorem (Abboud-Lewi-Williams'14)

k-Vector Sum Problem is W[1]-hard even when $h = O(k^2)$.

Lin's Proof Outline

Encoding Function $g : \mathbb{F}_2^{h \log n} \to \mathbb{F}_2^{2h \log n}$

Lin's Proof Outline

Encoding Function $g : \mathbb{F}_2^{h \log n} \to \mathbb{F}_2^{2h \log n}$

Special Subspace of $\mathbb{F}_2^{2h \log n}$: $\{\vec{\alpha} \circ \vec{\alpha} \cdots \circ \vec{\alpha} \mid \vec{\alpha} \in \mathbb{F}_2^{h}\}$

Lin's Proof Outline

Encoding Function $g : \mathbb{F}_2^{h \log n} \to \mathbb{F}_2^{2h \log n}$

Special Subspace of $\mathbb{F}_2^{2h \log n}$: $\{\vec{\alpha} \circ \vec{\alpha} \cdots \circ \vec{\alpha} \mid \vec{\alpha} \in \mathbb{F}_2^h\}$

$\langle \cdot \rangle : \mathbb{F}_2^h \times \mathbb{F}_2^{2h \log n} \to \mathbb{F}_2^{2 \log n}$

Lin's Proof Outline

$$\text{Encoding Function } g : \mathbb{F}_2^{h \log n} \to \mathbb{F}_2^{2h \log n}$$

$$\text{Special Subspace of } \mathbb{F}_2^{2h \log n}\text{: } \{\vec{\alpha} \circ \vec{\alpha} \cdots \circ \vec{\alpha} \mid \vec{\alpha} \in \mathbb{F}_2^{h}\}$$

$$\langle \cdot \rangle : \mathbb{F}_2^{h} \times \mathbb{F}_2^{2h \log n} \to \mathbb{F}_2^{2 \log n}$$

- Every linear combination of vectors in U is non-zero under g

Lin's Proof Outline

Encoding Function $g : \mathbb{F}_2^{h \log n} \to \mathbb{F}_2^{2h \log n}$

Special Subspace of $\mathbb{F}_2^{2h \log n}$: $\{\vec{\alpha} \circ \vec{\alpha} \cdots \circ \vec{\alpha} \mid \vec{\alpha} \in \mathbb{F}_2^h\}$

$\langle \cdot \rangle : \mathbb{F}_2^h \times \mathbb{F}_2^{2h \log n} \to \mathbb{F}_2^{2 \log n}$

- Every linear combination of vectors in U is non-zero under g
- For every $\vec{\alpha} \neq \vec{\beta} \in \mathbb{F}_2^h$ and $u, v, w \in \mathbb{F}_2^{h \log n}$, we have $\langle \vec{\alpha}, g(w+u) \rangle \neq \langle \vec{\beta}, g(w+v) \rangle \in \mathbb{F}_2^{2 \log n}$

Lin's Proof Outline

Construction of 3-CSP:

Lin's Proof Outline

Construction of 3-CSP:

Variables: $\mathbb{F}_2^{h\cdot k}$, Alphabet: $\mathbb{F}_2^{2\log n}$

Lin's Proof Outline

Construction of 3-CSP:

Variables: $\mathbb{F}_2^{h \cdot k}$, Alphabet: $\mathbb{F}_2^{2 \log n}$

Assignment: $(\vec{\alpha_1}, \ldots, \vec{\alpha_k}) \Longrightarrow \sum_{i \in [k]} \langle \vec{\alpha}_i, g(u_i) \rangle$

Lin's Proof Outline

Construction of 3-CSP:

Variables: $\mathbb{F}_2^{h \cdot k}$, Alphabet: $\mathbb{F}_2^{2 \log n}$

Assignment: $(\vec{\alpha_1}, \ldots, \vec{\alpha_k}) \Longrightarrow \sum_{i \in [k]} \langle \vec{\alpha}_i, g(u_i) \rangle$

Constraints:

Lin's Proof Outline

Construction of 3-CSP:

Variables: $\mathbb{F}_2^{h\cdot k}$, Alphabet: $\mathbb{F}_2^{2\log n}$

Assignment: $(\vec{\alpha_1},\ldots,\vec{\alpha_k}) \Longrightarrow \sum_{i\in[k]} \langle \vec{\alpha}_i, g(u_i)\rangle$

Constraints:

- Linearity Testing:
  $\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_k})) + \sigma((\vec{\beta_1},\ldots,\vec{\beta_k})) = \sigma((\vec{\alpha_1}+\vec{\beta_1},\ldots,\vec{\alpha_k}+\vec{\beta_k}))$

Lin's Proof Outline

Construction of 3-CSP:

Variables: $\mathbb{F}_2^{h\cdot k}$, Alphabet: $\mathbb{F}_2^{2\log n}$

Assignment: $(\vec{\alpha_1},\ldots,\vec{\alpha_k}) \Longrightarrow \sum_{i\in[k]} \langle \vec{\alpha}_i, g(u_i)\rangle$

Constraints:

- Linearity Testing:
  $\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_k}))+\sigma((\vec{\beta_1},\ldots,\vec{\beta_k}))=\sigma((\vec{\alpha_1}+\vec{\beta_1},\ldots,\vec{\alpha_k}+\vec{\beta_k}))$
- Membership Testing:
  $\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_i}+\vec{\alpha},\ldots,\vec{\alpha_k}))-\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_i},\ldots,\vec{\alpha_k}))=\langle\vec{\alpha},g(u)\rangle$ for some $u\in U_i$

Lin's Proof Outline

Construction of 3-CSP:

Variables: $\mathbb{F}_2^{h\cdot k}$, Alphabet: $\mathbb{F}_2^{2\log n}$

Assignment: $(\vec{\alpha_1},\ldots,\vec{\alpha_k}) \implies \sum_{i\in[k]} \langle \vec{\alpha}_i, g(u_i)\rangle$

Constraints:

- Linearity Testing:
  $\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_k})) + \sigma((\vec{\beta_1},\ldots,\vec{\beta_k})) = \sigma((\vec{\alpha_1}+\vec{\beta_1},\ldots,\vec{\alpha_k}+\vec{\beta_k}))$
- Membership Testing:
  $\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_i}+\vec{\alpha},\ldots,\vec{\alpha_k})) - \sigma((\vec{\alpha_1},\ldots,\vec{\alpha_i},\ldots,\vec{\alpha_k})) = \langle \vec{\alpha}, g(u)\rangle$ for some $u \in U_i$
- Zero Testing: $\sigma((\vec{\alpha_1},\ldots,\vec{\alpha_k})) = \sigma((\vec{\alpha_1}+\vec{\alpha},\ldots,\vec{\alpha_k}+\vec{\alpha}))$

Lin's Proof Outline

Construction of 3-CSP:

Constraints:

- Linearity Testing: $\sigma((\vec{\alpha_1},\dots,\vec{\alpha_k}))+\sigma((\vec{\beta_1},\dots,\vec{\beta_k}))=\sigma((\vec{\alpha_1}+\vec{\beta_1},\dots,\vec{\alpha_k}+\vec{\beta_k}))$
- Membership Testing: $\sigma((\vec{\alpha_1},\dots,\vec{\alpha_i}+\vec{\alpha},\dots,\vec{\alpha_k}))-\sigma((\vec{\alpha_1},\dots,\vec{\alpha_i},\dots,\vec{\alpha_k}))=\langle\vec{\alpha},g(u)\rangle$ for some $u\in U_i$
- Zero Testing: $\sigma((\vec{\alpha_1},\dots,\vec{\alpha_k}))=\sigma((\vec{\alpha_1}+\vec{\alpha},\dots,\vec{\alpha_k}+\vec{\alpha}))$

Construction of Graph:

Lin's Proof Outline

Construction of 3-CSP:

Constraints:

- Linearity Testing:
  $\sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_k})) + \sigma((\vec{\beta_1}, \ldots, \vec{\beta_k})) = \sigma((\vec{\alpha_1} + \vec{\beta_1}, \ldots, \vec{\alpha_k} + \vec{\beta_k}))$
- Membership Testing:
  $\sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_i} + \vec{\alpha}, \ldots, \vec{\alpha_k})) - \sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_i}, \ldots, \vec{\alpha_k})) = \langle \vec{\alpha}, g(u) \rangle$ for some $u \in U_i$
- Zero Testing: $\sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_k})) = \sigma((\vec{\alpha_1} + \vec{\alpha}, \ldots, \vec{\alpha_k} + \vec{\alpha}))$

Construction of Graph:

Vertices: $\mathbb{F}_2^{h\cdot k} \times \mathbb{F}_2^{h\cdot k} \times \mathbb{F}_2^{\ell} \times \mathbb{F}_2^{\ell}$,

Lin's Proof Outline

Construction of 3-CSP:

Constraints:

- Linearity Testing: $\sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_k})) + \sigma((\vec{\beta_1}, \ldots, \vec{\beta_k})) = \sigma((\vec{\alpha_1} + \vec{\beta_1}, \ldots, \vec{\alpha_k} + \vec{\beta_k}))$
- Membership Testing: $\sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_i} + \vec{\alpha}, \ldots, \vec{\alpha_k})) - \sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_i}, \ldots, \vec{\alpha_k})) = \langle \vec{\alpha}, g(u) \rangle$ for some $u \in U_i$
- Zero Testing: $\sigma((\vec{\alpha_1}, \ldots, \vec{\alpha_k})) = \sigma((\vec{\alpha_1} + \vec{\alpha}, \ldots, \vec{\alpha_k} + \vec{\alpha}))$

Construction of Graph:

Vertices: $\mathbb{F}_2^{h\cdot k} \times \mathbb{F}_2^{h\cdot k} \times \mathbb{F}_2^{\ell} \times \mathbb{F}_2^{\ell}$,

Edges: Constraints of Membership and Zero testing

Our Result: Better W[1]-hardness of approximation

Theorem (K-Khot'21)

Approximating k-Clique to any $k^{o(1)}$ factor is W[1]-hard.

Our Result: Better W[1]-hardness of approximation

Theorem (K-Khot'21)

Approximating k-Clique to any $k^{o(1)}$ factor is W[1]-hard.

- Start from k-Vector Sum problem over $\mathbb{F}_q$

Our Result: Better W[1]-hardness of approximation

Theorem (K-Khot'21)

Approximating k-Clique to any $k^{o(1)}$ factor is W[1]-hard.

- Start from k-Vector Sum problem over $\mathbb{F}_q$
- Use List Decoding of Hadamard Codes over $\mathbb{F}_q$

Our Result: Better W[1]-hardness of approximation

Theorem (K-Khot'21)

Approximating k-Clique to any $k^{o(1)}$ factor is W[1]-hard.

- Start from k-Vector Sum problem over $\mathbb{F}_q$
- Use List Decoding of Hadamard Codes over $\mathbb{F}_q$
- Careful Analysis as all arugments have "noise"

k-Clique

Theorem (Lin'21)

Approximating k-Clique to any $O(1)$ factor is W[1]-hard.

Theorem (K-Khot'21)

Approximating k-Clique to any $k^{o(1)}$ factor is W[1]-hard.

Part 3

High Level Remarks

The Big Prize

2-CSP:

Input: Graph $G(V, E)$, Alphabet Σ, Constraints $\{\pi_e \subseteq \Sigma \times \Sigma \mid e \in E\}$

Output: Assignment $\sigma : V \to \Sigma$ maximizing $\Pr_{(u,v)\in E}[(\sigma(u), \sigma(v)) \in \pi_{(u,v)}]$

Parameterized Inapproximability Hypothesis (PIH)

(Lokshtanov-Ramanujan-Saurabh-Zehavi'17)

There exists $\delta > 0$ such that 2-CSP on k vertices and alphabet size n is W[1]-hard to approximate to $(1 - \delta)$ factor

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique
  - k-Set Cover: Distance properties of Error Correcting Codes

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique
  - k-Set Cover: Distance properties of Error Correcting Codes
  - k-Set Intersection: Extremal Graphs, Random Algebraic Constructions

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique
  - k-Set Cover: Distance properties of Error Correcting Codes
  - k-Set Intersection: Extremal Graphs, Random Algebraic Constructions
  - k-Clique: Locally Decodable Codes, List Decodable Codes

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique
  - k-Set Cover: Distance properties of Error Correcting Codes
  - k-Set Intersection: Extremal Graphs, Random Algebraic Constructions
  - k-Clique: Locally Decodable Codes, List Decodable Codes
- Coping Mechanisms since we cannot prove PIH

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique
  - k-Set Cover: Distance properties of Error Correcting Codes
  - k-Set Intersection: Extremal Graphs, Random Algebraic Constructions
  - k-Clique: Locally Decodable Codes, List Decodable Codes
- Coping Mechanisms since we cannot prove PIH
  - Stark contrast with NP-world

Current Techniques are Coping Mechanisms

- Hardness of Approximation Results for k-Set Cover, k-Set Intersection, and k-Clique
- Each with a unique Gap-Creation technique
  - k-Set Cover: Distance properties of Error Correcting Codes
  - k-Set Intersection: Extremal Graphs, Random Algebraic Constructions
  - k-Clique: Locally Decodable Codes, List Decodable Codes
- Coping Mechanisms since we cannot prove PIH
  - Stark contrast with NP-world
- Proving PIH might lead to Unified Framework

PIH from ETH

Theorem (Lin-Ren-Sun-Wang'21)

If every constant approximation algorithm for k-Clique requires $n^{\omega(k/\log k)}$ time then PIH is true.

PIH from ETH

Theorem (Lin-Ren-Sun-Wang'21)

If every constant approximation algorithm for k-Clique requires $n^{\omega(k/\log k)}$ time then PIH is true.

- Lin'21, **K**-Khot'21: Every constant approximation algorithm for k-Clique requires $n^{(\log k)^{\Omega(1)}}$ time

PIH from ETH

Theorem (Lin-Ren-Sun-Wang'21)

If every constant approximation algorithm for k-Clique requires $n^{\omega(k/\log k)}$ time then PIH is true.

- Lin'21, **K**-Khot'21: Every constant approximation algorithm for k-Clique requires $n^{(\log k)^{\Omega(1)}}$ time
- Lin-Ren-Sun-Wang'21: Every constant approximation algorithm for k-Clique requires $n^{\Omega(\log k)}$ time

Thank you!