Lower bounds for the parameterized complexity of MINIMUM FILL-IN and other completion problems

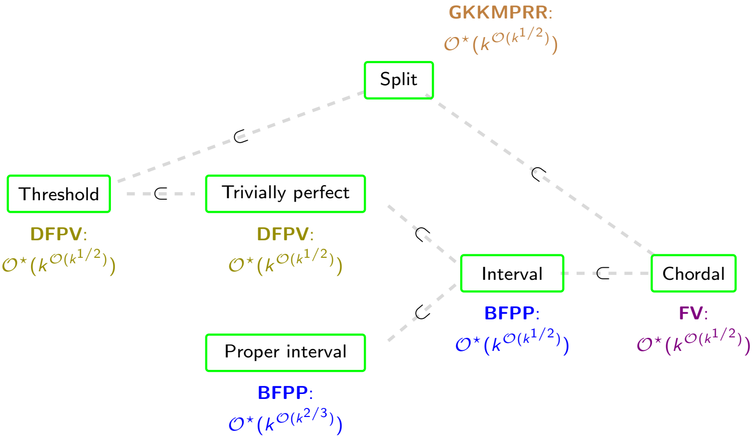
Michał Pilipczuk

Institute of Informatics, University of Warsaw

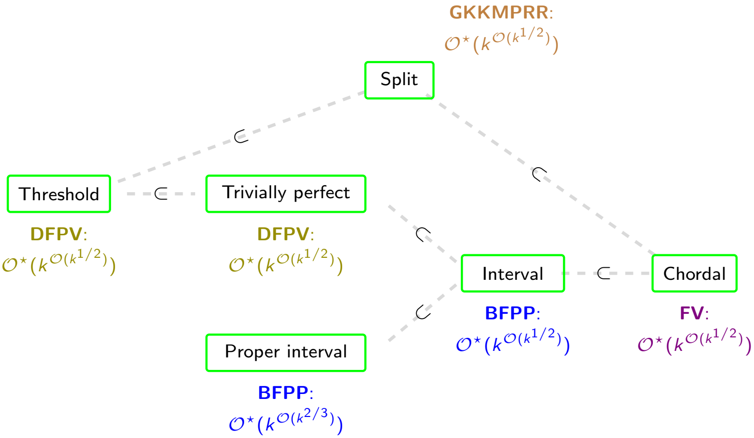
Joint work with Ivan Bliznets, Marek Cygan, Paweł Komosa and Lukáš Mach

Simons Institute,
November 4th, 2015

Motivation


GKKMPRR:
$\mathcal{O}^\star(k^{\mathcal{O}(k^{1/2})})$
Split
⊂
⊂
Threshold
⊂
Trivially perfect
⊂
Interval
⊂
Chordal
⊂
Proper interval
DFPV:
$\mathcal{O}^\star(k^{\mathcal{O}(k^{1/2})})$
DFPV:
$\mathcal{O}^\star(k^{\mathcal{O}(k^{1/2})})$
BFPP:
$\mathcal{O}^\star(k^{\mathcal{O}(k^{1/2})})$
FV:
$\mathcal{O}^\star(k^{\mathcal{O}(k^{1/2})})$
BFPP:
$\mathcal{O}^\star(k^{\mathcal{O}(k^{2/3})})$

Motivation



Is $\mathcal{O}^\star(2^{\tilde{\mathcal{O}}(k^{1/2})})$ the correct answer?

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:
  - An $\mathcal{O}^\star(2^{o(k^{1/2})})$ algorithm implies also $2^{o(n)}$.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:
  - An $\mathcal{O}^\star(2^{o(k^{1/2})})$ algorithm implies also $2^{o(n)}$.
  - Fundamental trade-off between cheap and expensive vertices.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:
  - An $\mathcal{O}^\star(2^{o(k^{1/2})})$ algorithm implies also $2^{o(n)}$.
  - Fundamental trade-off between cheap and expensive vertices.
- **Our answer**: We corroborate the suspicion that $k^{1/2}$ is optimum.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:
  - An $\mathcal{O}^\star(2^{o(k^{1/2})})$ algorithm implies also $2^{o(n)}$.
  - Fundamental trade-off between cheap and expensive vertices.
- **Our answer**: We corroborate the suspicion that $k^{1/2}$ is optimum.
  - **Note**: Ignore polylog factors.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:
  - An $\mathcal{O}^\star(2^{o(k^{1/2})})$ algorithm implies also $2^{o(n)}$.
  - Fundamental trade-off between cheap and expensive vertices.
- **Our answer**: We corroborate the suspicion that $k^{1/2}$ is optimum.
  - **Note**: Ignore polylog factors.
  - **Personal opinion**: $\log k$ in the exponent can be shaved off.

The square root phenomenon

- **Planar graphs**: Known NP-hardness reduction usually give $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bounds under ETH.
- **Completion problems**:
  - Known reductions give $\mathcal{O}^\star(2^{o(n^{1/6})})$ and $\mathcal{O}^\star(2^{o(k^{1/9})})$ lower bounds, or worse.
  - **Fomin & Villanger**:
    Here the big gap between what we suspect and what we know is frustrating.
- **Arguments for the optimality of $k^{1/2}$**:
  - An $\mathcal{O}^\star(2^{o(k^{1/2})})$ algorithm implies also $2^{o(n)}$.
  - Fundamental trade-off between cheap and expensive vertices.
- **Our answer**: We corroborate the suspicion that $k^{1/2}$ is optimum.
  - **Note**: Ignore polylog factors.
  - **Personal opinion**: $\log k$ in the exponent can be shaved off.
- **Goal**: Prove a $2^{o(n)}$ lower bound for Minimum Fill-in.

Results

- **Under ETH**:

Results

- **Under ETH**:
  - no $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ algorithm for Minimum Fill-in;

Results

- **Under ETH**:
  - no $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ algorithm for Minimum Fill-in;
  - consequently, no $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ FPT algorithm.

Results

- **Under ETH**:
  - no $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ algorithm for MINIMUM FILL-IN;
  - consequently, no $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ FPT algorithm.
- **Under stronger assumptions**:

Results

- **Under ETH**:
  - no $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ algorithm for Minimum Fill-in;
  - consequently, no $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ FPT algorithm.
- **Under stronger assumptions**:
  - no $2^{o(n)}$ algorithm for Minimum Fill-in;

Results

- **Under ETH**:
  - no $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ algorithm for Minimum Fill-in;
  - consequently, no $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ FPT algorithm.
- **Under stronger assumptions**:
  - no $2^{o(n)}$ algorithm for Minimum Fill-in;
  - consequently, no $\mathcal{O}^\star(2^{o(k^{1/2})})$ FPT algorithm.

Results

- **Under ETH**:
  - no $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ algorithm for Minimum Fill-in;
  - consequently, no $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ FPT algorithm.
- **Under stronger assumptions**:
  - no $2^{o(n)}$ algorithm for Minimum Fill-in;
  - consequently, no $\mathcal{O}^\star(2^{o(k^{1/2})})$ FPT algorithm.
- Same lower bounds for all the other completion problems.

Known reductions

3SAT ⟶ MaxCut ⟶ OLA ⟶ Fill-in

Known reductions

$$\text{3SAT} \longrightarrow \textsc{MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(n+m)$

$m' = \mathcal{O}(n+m)$

Known reductions

3SAT ⟶ MaxCut ⟶ OLA ⟶ Fill-in

$n' = \mathcal{O}(n + m)$

$m' = \mathcal{O}(n + m)$

$n'' = \mathcal{O}((n')^3)$

$m'' = \mathcal{O}((n')^6)$

Known reductions

$$\text{3SAT} \longrightarrow \textsc{MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(n + m)$ $\quad$ $n'' = \mathcal{O}((n')^3)$ $\quad$ $n''' = \mathcal{O}(\Delta \cdot n'')$

$m' = \mathcal{O}(n + m)$ $\quad$ $m'' = \mathcal{O}((n')^6)$

Known reductions

$$\text{3SAT} \longrightarrow \text{MaxCut} \longrightarrow \text{OLA} \longrightarrow \text{Fill-in}$$

$n' = \mathcal{O}(n+m)$

$m' = \mathcal{O}(n+m)$

$n'' = \mathcal{O}((n')^3)$

$m'' = \mathcal{O}((n')^6)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- Let $\pi\colon V(G) \to \{1, 2, \ldots, n\}$ be an ordering of $V(G)$.

Optimum Linear Arrangement

- Let $\pi\colon V(G) \to \{1, 2, \ldots, n\}$ be an ordering of $V(G)$.
- The *cost* of an edge uv is $c(uv) = |\pi(u) - \pi(v)|$

Optimum Linear Arrangement

- Let $\pi \colon V(G) \to \{1, 2, \ldots, n\}$ be an ordering of $V(G)$.
- The *cost* of an edge uv is $c(uv) = |\pi(u) - \pi(v)|$
- The *cost* of π is $\sum_{e \in E(G)} c(e)$.

Optimum Linear Arrangement

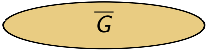
- Let $\pi\colon V(G) \to \{1, 2, \ldots, n\}$ be an ordering of $V(G)$.
- The *cost* of an edge uv is $c(uv) = |\pi(u) - \pi(v)|$
- The *cost* of π is $\sum_{e \in E(G)} c(e)$.
- Optimum Linear Arrangement: Find π with minimum cost.

Optimum Linear Arrangement

- Let $\pi\colon V(G) \to \{1, 2, \dots, n\}$ be an ordering of $V(G)$.
- The *cost* of an edge uv is $c(uv) = |\pi(u) - \pi(v)|$
- The *cost* of π is $\sum_{e \in E(G)} c(e)$.
- Optimum Linear Arrangement: Find π with minimum cost.
- **Note**: It can be as large as cubic.

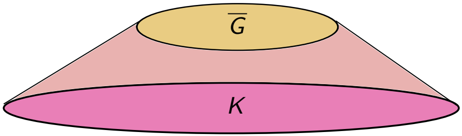
Reduction MaxCut⇝OLA

- Complement the graph.



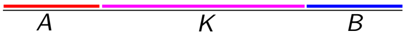
Reduction MaxCut⇝OLA

- Complement the graph.
- Add a clique K of size $N = n^c$ for a large c, and make it fully adjacent to the rest of the graph.



Reduction MaxCut⇝OLA

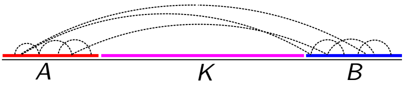
- Complement the graph.
- Add a clique K of size $N = n^c$ for a large c, and make it fully adjacent to the rest of the graph.



- **Easy**: K can be assumed to be consecutive.

Reduction MaxCut⇝OLA

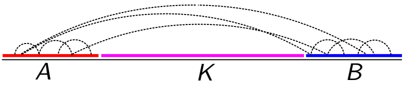
- Complement the graph.
- Add a clique K of size $N = n^c$ for a large c, and make it fully adjacent to the rest of the graph.



- **Easy**: K can be assumed to be consecutive.
- Instead of minimizing the cost, maximize the cost of the non-edges.

Reduction MaxCut$\rightsquigarrow$OLA

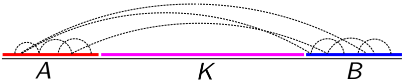
- Complement the graph.
- Add a clique K of size $N = n^c$ for a large c, and make it fully adjacent to the rest of the graph.



- **Easy**: K can be assumed to be consecutive.
- Instead of minimizing the cost, maximize the cost of the non-edges.
- We want to maximize the number of non-edges flying over K.

Reduction MaxCut⇝OLA

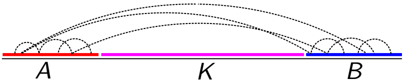
- Complement the graph.
- Add a clique K of size $N = n^c$ for a large c, and make it fully adjacent to the rest of the graph.



- **Easy**: K can be assumed to be consecutive.
- Instead of minimizing the cost, maximize the cost of the non-edges.
- We want to maximize the number of non-edges flying over K.
- Every edge flying over K has to gain more than the total noise on the sides.

Reduction MaxCut⇝OLA

- Complement the graph.
- Add a clique K of size $N = n^c$ for a large c, and make it fully adjacent to the rest of the graph.



- **Easy**: K can be assumed to be consecutive.
- Instead of minimizing the cost, maximize the cost of the non-edges.
- We want to maximize the number of non-edges flying over K.
- Every edge flying over K has to gain more than the total noise on the sides.
  - Hence K must be large to make this work.

Idea

- Start with an instance that has a **gap**.

Idea

- Start with an instance that has a **gap**.
- $\textsc{Gap MaxCut}_{[\alpha,\beta]}$: distinguish between $OPT \leq \alpha m$ and $OPT \geq \beta m$.

Idea

- Start with an instance that has a **gap**.
- $\textsc{Gap MaxCut}_{[\alpha,\beta]}$: distinguish between $OPT \leq \alpha m$ and $OPT \geq \beta m$.
- Suppose $\textsc{Gap MaxCut}_{[\alpha,\beta]}$ is hard for some $0 \leq \alpha < \beta \leq 1$.

Idea

- Start with an instance that has a **gap**.
- $\textsc{Gap MaxCut}_{[\alpha,\beta]}$: distinguish between $OPT \leq \alpha m$ and $OPT \geq \beta m$.
- Suppose $\textsc{Gap MaxCut}_{[\alpha,\beta]}$ is hard for some $0 \leq \alpha < \beta \leq 1$.
- Then set $|K| = \lceil \frac{2}{\beta-\alpha} \rceil \cdot n$.

Idea

- Start with an instance that has a **gap**.
- $\textsc{Gap MaxCut}_{[\alpha,\beta]}$: distinguish between $OPT \leq \alpha m$ and $OPT \geq \beta m$.
- Suppose $\textsc{Gap MaxCut}_{[\alpha,\beta]}$ is hard for some $0 \leq \alpha < \beta \leq 1$.
- Then set $|K| = \lceil \frac{2}{\beta - \alpha} \rceil \cdot n$.
- Gap in $\textsc{MaxCut} \rightsquigarrow$ Gap of $\geq 2nm$ on edges flying over K.

Idea

- Start with an instance that has a **gap**.
- $\textsc{Gap MaxCut}_{[\alpha,\beta]}$: distinguish between $OPT \leq \alpha m$ and $OPT \geq \beta m$.
- Suppose $\textsc{Gap MaxCut}_{[\alpha,\beta]}$ is hard for some $0 \leq \alpha < \beta \leq 1$.
- Then set $|K| = \lceil \frac{2}{\beta - \alpha} \rceil \cdot n$.
- Gap in $\textsc{MaxCut} \rightsquigarrow$ Gap of $\geq 2nm$ on edges flying over K.
- Maximum noise is smaller than nm, so the gap amortizes the noise.

Hardness of Gap MaxCut

- **Sparsification lemma** $\Rightarrow 2^{o(m)}$ hardness of 3SAT

Hardness of GAP MAXCUT

- **Sparsification lemma** $\Rightarrow 2^{o(m)}$ hardness of $\mathrm{3SAT}$
- **Almost linear PCPs**: Reduction from $\mathrm{3SAT}$ to GAP $\mathrm{3SAT}_{[r,1]}$ that increases the size by $\log^c m$.

Hardness of GAP MAXCUT

- **Sparsification lemma** $\Rightarrow 2^{o(m)}$ hardness of $3\mathrm{SAT}$
- **Almost linear PCPs**: Reduction from $3\mathrm{SAT}$ to GAP $3\mathrm{SAT}_{[r,1]}$ that increases the size by $\log^c m$.

Corollary

Under ETH, there exists constants $r < 1$ and c such that there is no $2^{\mathcal{O}(m/\log^c m)}$ algorithm for GAP $3\mathrm{SAT}_{[r,1]}$.

Hardness of Gap MaxCut

- **Sparsification lemma** $\Rightarrow 2^{o(m)}$ hardness of 3SAT
- **Almost linear PCPs**: Reduction from 3SAT to Gap $3\mathrm{SAT}_{[r,1]}$ that increases the size by $\log^c m$.

Corollary

Under ETH, there exists constants $r < 1$ and c such that there is no $2^{\mathcal{O}(m/\log^c m)}$ algorithm for Gap $3\mathrm{SAT}_{[r,1]}$.

- Already observed and used by Marx in 2007 for proving lower bounds on the running times of geometric PTASes.

Hardness of GAP MAXCUT

- **Sparsification lemma** $\Rightarrow 2^{o(m)}$ hardness of 3SAT
- **Almost linear PCPs**: Reduction from 3SAT to GAP $3\mathrm{SAT}_{[r,1]}$ that increases the size by $\log^c m$.

Corollary

Under ETH, there exists constants $r < 1$ and c such that there is no $2^{\mathcal{O}(m/\log^c m)}$ algorithm for GAP $3\mathrm{SAT}_{[r,1]}$.

- Already observed and used by Marx in 2007 for proving lower bounds on the running times of geometric PTASes.
- Standard reductions to MAXCUT preserve the gap $\Rightarrow$ $2^{\mathcal{O}(m/\log^c m)}$ hardness of GAP $\mathrm{MAXCUT}_{[\alpha,\beta]}$.

Hardness of Gap MaxCut

- **Sparsification lemma** $\Rightarrow 2^{o(m)}$ hardness of $3\mathrm{SAT}$
- **Almost linear PCPs**: Reduction from $3\mathrm{SAT}$ to Gap $3\mathrm{SAT}_{[r,1]}$ that increases the size by $\log^c m$.

Corollary

Under ETH, there exists constants $r < 1$ and c such that there is no $2^{\mathcal{O}(m/\log^c m)}$ algorithm for Gap $3\mathrm{SAT}_{[r,1]}$.

- Already observed and used by Marx in 2007 for proving lower bounds on the running times of geometric PTASes.
- Standard reductions to MaxCut preserve the gap $\Rightarrow$ $2^{\mathcal{O}(m/\log^c m)}$ hardness of Gap $\mathrm{MaxCut}_{[\alpha,\beta]}$.
- **Cor**: Under ETH, there is no $2^{\mathcal{O}(n/\log^c n)}$ algorithm for OLA, for some c.

Reductions

$$\text{3SAT} \longrightarrow \text{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \text{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

Reductions

$$\text{3SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

Reductions

$$3\text{SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- We obtained a $2^{\mathcal{O}(n/\log^c n)}$ lower bound for OLA, but not on sparse graphs.

Reductions

$$\text{3SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- We obtained a $2^{\mathcal{O}(n/\log^c n)}$ lower bound for OLA, but not on sparse graphs.
- This proves $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ lower bound for Minimum Fill-in.

Reductions

$$\text{3SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- We obtained a $2^{\mathcal{O}(n/\log^c n)}$ lower bound for OLA, but not on sparse graphs.
- This proves $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ lower bound for Minimum Fill-in.
- Two routes to rescue the situation:

Reductions

$$\text{3SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- We obtained a $2^{\mathcal{O}(n/\log^c n)}$ lower bound for OLA, but not on sparse graphs.
- This proves $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ lower bound for MINIMUM FILL-IN.
- Two routes to rescue the situation:
  - **Plan 1**: Show hardness of OLA on bounded degree graphs.

Reductions

$$\text{3SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- We obtained a $2^{\mathcal{O}(n/\log^c n)}$ lower bound for OLA, but not on sparse graphs.
- This proves $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ lower bound for Minimum Fill-in.
- Two routes to rescue the situation:
  - **Plan 1**: Show hardness of OLA on bounded degree graphs.
  - **Plan 2**: Find a better reduction from OLA to Minimum Fill-in.

Reductions

$$\text{3SAT} \longrightarrow \textsc{Gap MaxCut} \longrightarrow \text{OLA} \longrightarrow \textsc{Fill-in}$$

$n' = \mathcal{O}(m \log^c m)$

$m' = \mathcal{O}(m \log^c m)$

$n'' = \mathcal{O}(n')$

$m'' = \mathcal{O}((n')^2)$

$n''' = \mathcal{O}(\Delta \cdot n'')$

- We obtained a $2^{\mathcal{O}(n/\log^c n)}$ lower bound for OLA, but not on sparse graphs.
- This proves $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ lower bound for MINIMUM FILL-IN.
- Two routes to rescue the situation:
  - **Plan 1**: Show hardness of OLA on bounded degree graphs.
  - **Plan 2**: Find a better reduction from OLA to MINIMUM FILL-IN.
- Let's look at the reduction OLA⇝MINIMUM FILL-IN first.

Reduction of Yannakakis

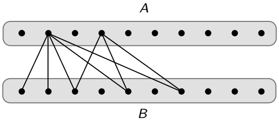
- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.

Reduction of Yannakakis

- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.

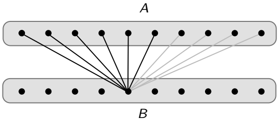
Reduction of Yannakakis

- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.
  - Eq., there is an ordering π of A such that $N(u) \supseteq N(v)$ for $\pi(u) \leq \pi(v)$.



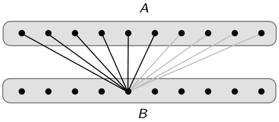
Reduction of Yannakakis

- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.
  - Eq., there is an ordering π of A such that $N(u) \supseteq N(v)$ for $\pi(u) \leq \pi(v)$.
  - Eq., neighborhoods of vertices of B are downward closed w.r.t. π.



Reduction of Yannakakis

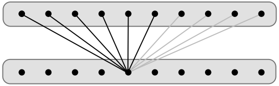
- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.
  - Eq., there is an ordering π of A such that $N(u) \supseteq N(v)$ for $\pi(u) \leq \pi(v)$.
  - Eq., neighborhoods of vertices of B are downward closed w.r.t. π.



- CHAIN COMPLETION: Add at most k edges to a given bipartite graph with a fixed bipartition to obtain a chain graph.

Reduction of Yannakakis

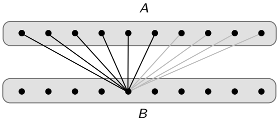
- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.
  - Eq., there is an ordering π of A such that $N(u) \supseteq N(v)$ for $\pi(u) \leq \pi(v)$.
  - Eq., neighborhoods of vertices of B are downward closed w.r.t. π.



- Chain Completion: Add at most k edges to a given bipartite graph with a fixed bipartition to obtain a chain graph.
  - Chain Completion $\rightsquigarrow$ Chordal/Interval/Proper Interval Completion: Make both A and B into cliques.

Reduction of Yannakakis

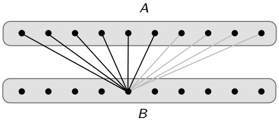
- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.
  - Eq., there is an ordering π of A such that $N(u) \supseteq N(v)$ for $\pi(u) \le \pi(v)$.
  - Eq., neighborhoods of vertices of B are downward closed w.r.t. π.



- CHAIN COMPLETION: Add at most k edges to a given bipartite graph with a fixed bipartition to obtain a chain graph.
  - CHAIN COMPLETION$\rightsquigarrow$CHORDAL/INTERVAL/PROPER INTERVAL COMPLETION: Make both A and B into cliques.
  - CHAIN COMPLETION$\rightsquigarrow$THRESHOLD/TRIVIALLY PERFECT COMPLETION: Make A into a clique.

Reduction of Yannakakis

- **Def.** A bipartite graph (A, B, E) is a *chain graph* if $\{N(u)\}_{u \in A}$ form a chain in the inclusion order.
  - Eq., this holds for $\{N(v)\}_{v \in B}$.
  - Eq., there is an ordering π of A such that $N(u) \supseteq N(v)$ for $\pi(u) \leq \pi(v)$.
  - Eq., neighborhoods of vertices of B are downward closed w.r.t. π.



- Chain Completion: Add at most k edges to a given bipartite graph with a fixed bipartition to obtain a chain graph.
  - Chain Completion⇝Chordal/Interval/Proper Interval Completion: Make both A and B into cliques.
  - Chain Completion⇝Threshold/Trivially Perfect Completion: Make A into a clique.
- **Cor**: Suffices to get reduction OLA⇝Chain Completion

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.
- Fix some ordering $\pi \colon V(G) \to \{1, 2, \ldots, n\}$, and suppose this is the target ordering of neighborhoods.

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.
- Fix some ordering $\pi \colon V(G) \to \{1, 2, \ldots, n\}$, and suppose this is the target ordering of neighborhoods.
- For every $uv \in E(G) = B$, we need to add $\max(\pi(u), \pi(v)) - 2$ edges.

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.
- Fix some ordering $\pi\colon V(G) \to \{1, 2, \ldots, n\}$, and suppose this is the target ordering of neighborhoods.
- For every $uv \in E(G) = B$, we need to add $\max(\pi(u), \pi(v)) - 2$ edges.
- **Crucial observation**:

$$2 \cdot \max(\pi(u), \pi(v)) = (\pi(u) + \pi(v)) + |\pi(u) - \pi(v)|$$

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.
- Fix some ordering $\pi\colon V(G) \to \{1, 2, \ldots, n\}$, and suppose this is the target ordering of neighborhoods.
- For every $uv \in E(G) = B$, we need to add $\max(\pi(u), \pi(v)) - 2$ edges.
- **Crucial observation**:

$$2 \cdot \max(\pi(u), \pi(v)) = (\pi(u) + \pi(v)) + |\pi(u) - \pi(v)|$$

- The sum of $\pi(u) + \pi(v)$ summands is constant if the input graph is regular.

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.
- Fix some ordering $\pi\colon V(G) \to \{1, 2, \ldots, n\}$, and suppose this is the target ordering of neighborhoods.
- For every $uv \in E(G) = B$, we need to add $\max(\pi(u), \pi(v)) - 2$ edges.
- **Crucial observation**:

$$2 \cdot \max(\pi(u), \pi(v)) = (\pi(u) + \pi(v)) + |\pi(u) - \pi(v)|$$

- The sum of $\pi(u) + \pi(v)$ summands is constant if the input graph is regular.
- Can be easily achieved by adding loops.

Reduction of Yannakakis

- Set $A = V(G)$ and $B = E(G)$.
- Connect $u \in A$ with $e \in B$ iff u is an endpoint of e.
- Fix some ordering $\pi\colon V(G) \to \{1, 2, \ldots, n\}$, and suppose this is the target ordering of neighborhoods.
- For every $uv \in E(G) = B$, we need to add $\max(\pi(u), \pi(v)) - 2$ edges.
- **Crucial observation**:

$$2 \cdot \max(\pi(u), \pi(v)) = (\pi(u) + \pi(v)) + |\pi(u) - \pi(v)|$$

- The sum of $\pi(u) + \pi(v)$ summands is constant if the input graph is regular.
- Can be easily achieved by adding loops.
- **Ergo**: Minimization of the number of fill edges is equivalent to minimization of the OLA cost.

Issues

- **Wanted**: $2^{o(n)}$ hardness for OLA on bounded degree graphs.

Issues

- **Wanted**: $2^{o(n)}$ hardness for OLA on bounded degree graphs.
- Route via MaxCut: We would need hardness of MaxCut on co-bounded degree graphs.

Issues

- **Wanted**: $2^{o(n)}$ hardness for OLA on bounded degree graphs.
- Route via MaxCut: We would need hardness of MaxCut on co-bounded degree graphs.
- **Our approach**:

Issues

- **Wanted**: $2^{o(n)}$ hardness for OLA on bounded degree graphs.
- Route via MaxCut: We would need hardness of MaxCut on co-bounded degree graphs.
- **Our approach**:
  - Introduce a new hypothesis about approximability of Minimum Bisection.

Issues

- **Wanted**: $2^{o(n)}$ hardness for OLA on bounded degree graphs.
- Route via MaxCut: We would need hardness of MaxCut on co-bounded degree graphs.
- **Our approach**:
  - Introduce a new hypothesis about approximability of Minimum Bisection.
  - Prove that starting with this hypothesis we can make this plan work.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:
  - NP-hard and APX-hard on bounded degree graphs.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:
  - NP-hard and APX-hard on bounded degree graphs.
  - The best known approximation has apx factor $\mathcal{O}(\log OPT)$.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:
  - NP-hard and APX-hard on bounded degree graphs.
  - The best known approximation has apx factor $\mathcal{O}(\log OPT)$.
  - The problem is FPT.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:
  - NP-hard and APX-hard on bounded degree graphs.
  - The best known approximation has apx factor $\mathcal{O}(\log OPT)$.
  - The problem is FPT.
  - No $2^{o(n)}$ lower bound on bounded degree graphs.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:
  - NP-hard and APX-hard on bounded degree graphs.
  - The best known approximation has apx factor $\mathcal{O}(\log OPT)$.
  - The problem is FPT.
  - No $2^{o(n)}$ lower bound on bounded degree graphs.

Hypothesis

There exist $0 \leq \alpha < \beta \leq 1$ and $d \in \mathbb{N}$ such that there is no $2^{o(n)}$-time algorithm for Gap MinBisection$_{[\alpha,\beta]}$ on d-regular graphs.

Hypothesis

- MinBisection: Partition the vertex set into equal halves so that the number of edges crossing the partition is minimized.
- What is known:
  - NP-hard and APX-hard on bounded degree graphs.
  - The best known approximation has apx factor $\mathcal{O}(\log OPT)$.
  - The problem is FPT.
  - No $2^{o(n)}$ lower bound on bounded degree graphs.

Hypothesis

There exist $0 \leq \alpha < \beta \leq 1$ and $d \in \mathbb{N}$ such that there is no $2^{o(n)}$-time algorithm for Gap MinBisection$_{[\alpha,\beta]}$ on d-regular graphs.

- **Intuition**: MinBisection on bounded degree graphs does not admit a subexponential-time approximation scheme.

Reduction $\textsc{MinBisection} \rightsquigarrow \textrm{OLA}$

- **First attempt**:

Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.

Reduction MinBisection$\rightsquigarrow$OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.

Reduction MinBisection$\rightsquigarrow$OLA

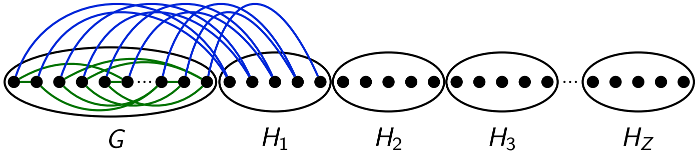
- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?

Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?
- **Second attempt**:

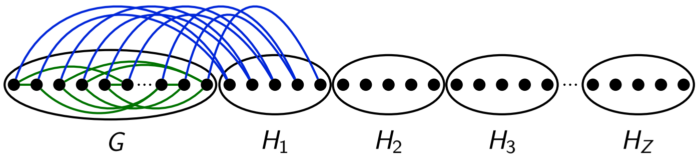
Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?
- **Second attempt**:
  - Replace K with a careful construction consisting of a constant-length chain of expanders of increasing degrees.



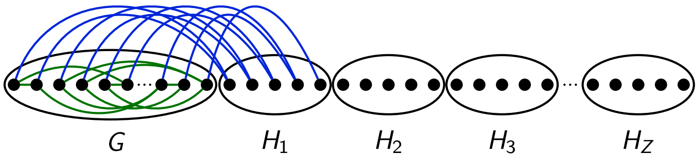
Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?
- **Second attempt**:
  - Replace K with a careful construction consisting of a constant-length chain of expanders of increasing degrees.
  - The chain behaves in a rigid manner.



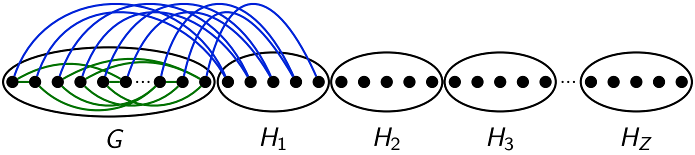
Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?
- **Second attempt**:
  - Replace K with a careful construction consisting of a constant-length chain of expanders of increasing degrees.
  - The chain behaves in a rigid manner.
  - Replace the full join with 'balanced' connections between expanders and G.



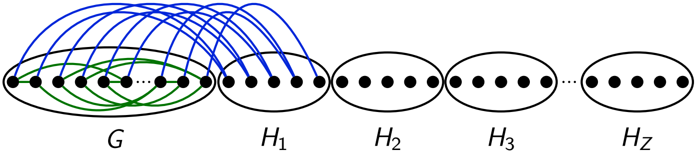
Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?
- **Second attempt**:
  - Replace K with a careful construction consisting of a constant-length chain of expanders of increasing degrees.
  - The chain behaves in a rigid manner.
  - Replace the full join with 'balanced' connections between expanders and G.
  - The neighborhoods of G are 'uniformly distributed' in K.



Reduction MinBisection⇝OLA

- **First attempt**:
  - Start with a hard instance G of Gap MinBisection.
  - Replace K with a large constant-degree expander.
  - What to do with the full join between K and the rest of the graph?
- **Second attempt**:
  - Replace K with a careful construction consisting of a constant-length chain of expanders of increasing degrees.
  - The chain behaves in a rigid manner.
  - Replace the full join with 'balanced' connections between expanders and G.
  - The neighborhoods of G are 'uniformly distributed' in K.
  - Do the maths to make sure that the gap swallows the possible noise.



Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4}/\log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4}/\log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.
- **Open problems**:

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.
- **Open problems**:
  - Investigate the Hypothesis.

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.
- **Open problems**:
  - Investigate the Hypothesis.
  - Equivalence of OLA hardness and the Hypothesis?

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.
- **Open problems**:
  - Investigate the Hypothesis.
  - Equivalence of OLA hardness and the Hypothesis?
  - Different route to hardness of Minimum Fill-in?

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2}/\log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4}/\log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.
- **Open problems**:
  - Investigate the Hypothesis.
  - Equivalence of OLA hardness and the Hypothesis?
  - Different route to hardness of Minimum Fill-in?
  - Tight lower bounds for Feedback Arc Set in Tournaments?

Conclusions

- We proved suboptimal lower bounds under ETH and almost tight lower bounds under the Hypothesis.
- **Message**: Look at approximation hardness for MinBisection first.
- Approach can be also used to get $2^{\mathcal{O}(n^{1/2} / \log^c n)}$ and $\mathcal{O}^\star(2^{\mathcal{O}(k^{1/4} / \log^c k)})$ lower bounds for Feedback Arc Set in Tournaments.
- **Open problems**:
  - Investigate the Hypothesis.
  - Equivalence of OLA hardness and the Hypothesis?
  - Different route to hardness of Minimum Fill-in?
  - Tight lower bounds for Feedback Arc Set in Tournaments?
- **Thanks for your attention!**