Tight bounds for parameterized complexity of Cluster Editing

Fedor V. Fomin[1], Stefan Kratsch[2], Marcin Pilipczuk[3], Michał Pilipczuk[1], Yngve Villanger[1]

[1] Department of Informatics, University of Bergen, Norway

[2] Technische Universität Berlin, Germany

[3] Institute of Informatics, University of Warsaw, Poland

STACS 2013, February 28th

Problem

Cluster Editing

Problem

Cluster Editing

- **Input:** Graph G, integer k

Problem

Cluster Editing

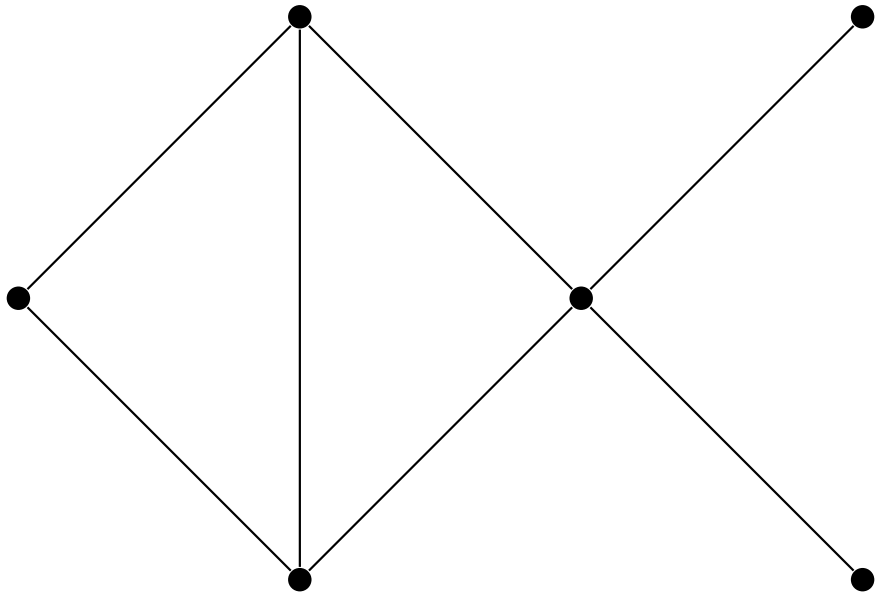
- **Input:** Graph G, integer k
- **Parameter:** k

Problem

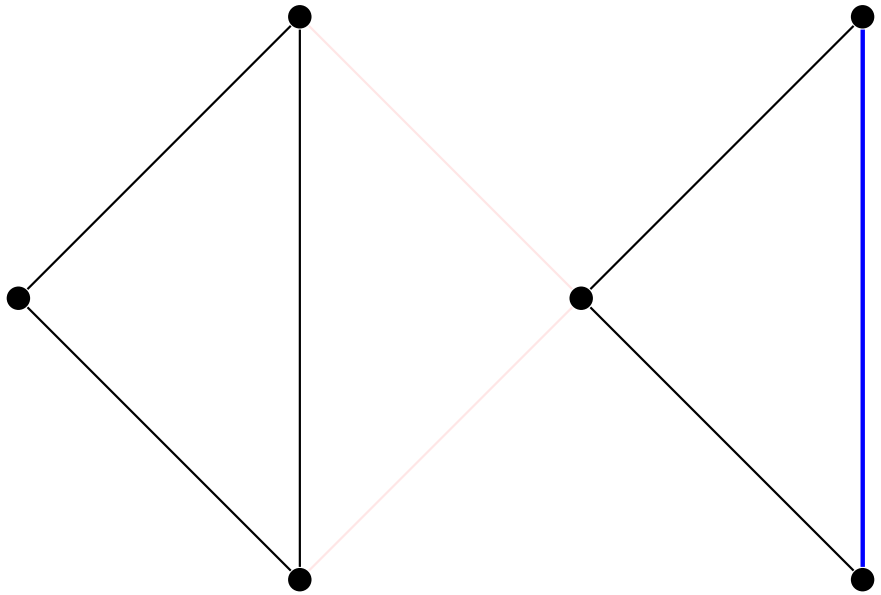
Cluster Editing

- **Input:** Graph G, integer k
- **Parameter:** k
- **Question:** Can one make at most k edge additions/deletions to turn G into a cluster graph, i.e., a disjoint union of cliques?

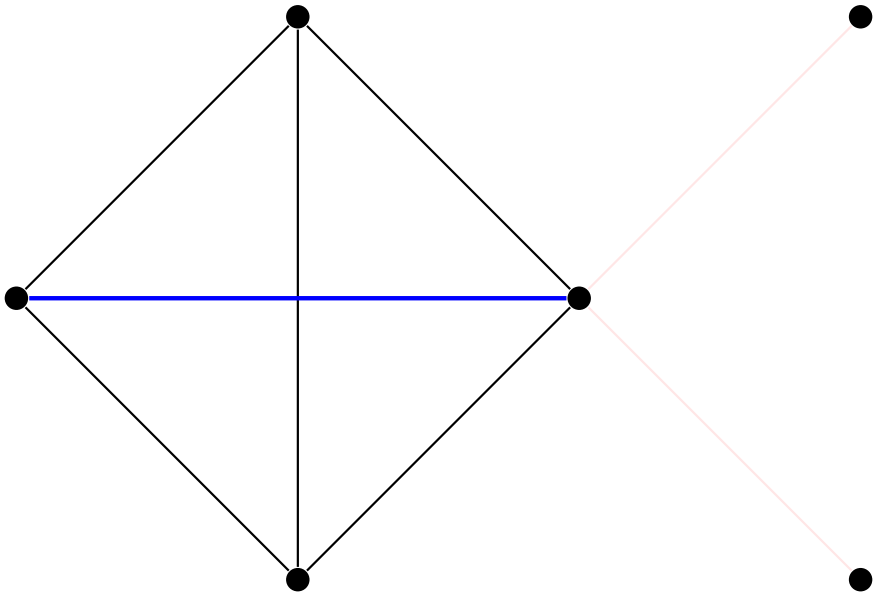
Example



Example



Example



Why Cluster Editing?

- Enormous potential for applications:

Why Cluster Editing?

- Enormous potential for applications:
  - computational biology;

Why Cluster Editing?

- Enormous potential for applications:
  - computational biology;
  - machine learning.

Why Cluster Editing?

- Enormous potential for applications:
  - computational biology;
  - machine learning.
- **Reason**: Models the natural problem of clustering objects, where a simple similarity relation is given.

What is known?

- **Classical complexity**: *NP*-complete

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:
  - 3-approximation [Bansal, Blum, Chawla],

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:
  - 3-approximation [Bansal, Blum, Chawla],
  - but APX-hard [Charikar, Guruswami, Wirth].

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:
  - 3-approximation [Bansal, Blum, Chawla],
  - but APX-hard [Charikar, Guruswami, Wirth].
- **Parameterized complexity**:

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:
  - 3-approximation [Bansal, Blum, Chawla],
  - but APX-hard [Charikar, Guruswami, Wirth].
- **Parameterized complexity**:
  - $\mathcal{O}^\star(1.62^k)$ FPT algorithm [Böcker],

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:
  - 3-approximation [Bansal, Blum, Chawla],
  - but APX-hard [Charikar, Guruswami, Wirth].
- **Parameterized complexity**:
  - $\mathcal{O}^\star(1.62^k)$ FPT algorithm [Böcker],
  - kernel with at most $2k$ vertices [Cao, Chen, Meng],

What is known?

- **Classical complexity**: *NP*-complete
- **Approximation**:
  - 3-approximation [Bansal, Blum, Chawla],
  - but APX-hard [Charikar, Guruswami, Wirth].
- **Parameterized complexity**:
  - $\mathcal{O}^\star(1.62^k)$ FPT algorithm [Böcker],
  - kernel with at most $2k$ vertices [Cao, Chen, Meng],
  - no $\mathcal{O}^\star(2^{o(k)})$ algorithm under ETH [this work; Komusiewicz].

Bounding the target number of clusters

- Although the general problem is APX-hard...

Bounding the target number of clusters

- Although the general problem is APX-hard...
- There is a PTAS, if we ask for a cluster graph with at most p clusters, where p is a constant! [Giotis, Guruswami]

Bounding the target number of clusters

- Although the general problem is APX-hard...
- There is a PTAS, if we ask for a cluster graph with at most p clusters, where p is a constant! [Giotis, Guruswami]
- **Running time**: $n^{\mathcal{O}(9^p/\epsilon^2)}$

Bounding the target number of clusters

- Although the general problem is APX-hard...
- There is a PTAS, if we ask for a cluster graph with at most p clusters, where p is a constant! [Giotis, Guruswami]
- **Running time**: $n^{\mathcal{O}(9^p/\epsilon^2)}$
- **Our goal**: Study parameterized complexity of Cluster Editing, when the target number of clusters is bounded by a sublinear function of k.

Problem definition(s)

p-Cluster Editing

- **Input:** Graph G, integers p, k
- **Parameters:** p, k
- **Question:** Can one make at most k edge additions/deletions to turn G into a cluster graph with **exactly** p clusters?

Problem definition(s)

$p_{\leqslant}$-Cluster Editing

- **Input:** Graph G, integers p, k
- **Parameters:** p, k
- **Question:** Can one make at most k edge additions/deletions to turn G into a cluster graph with **at most** p clusters?

Problem definition(s)

$p_{\leqslant}$-Cluster Editing

- **Input:** Graph G, integers p, k
- **Parameters:** p, k
- **Question:** Can one make at most k edge additions/deletions to turn G into a cluster graph with **at most** p clusters?

- p-Cluster Editing is at least as hard as $p_{\leqslant}$-Cluster Editing, but the converse is unclear.

Problem definition(s)

$p_{\leqslant}$-Cluster Editing

- **Input:** Graph G, integers p, k
- **Parameters:** p, k
- **Question:** Can one make at most k edge additions/deletions to turn G into a cluster graph with **at most** p clusters?

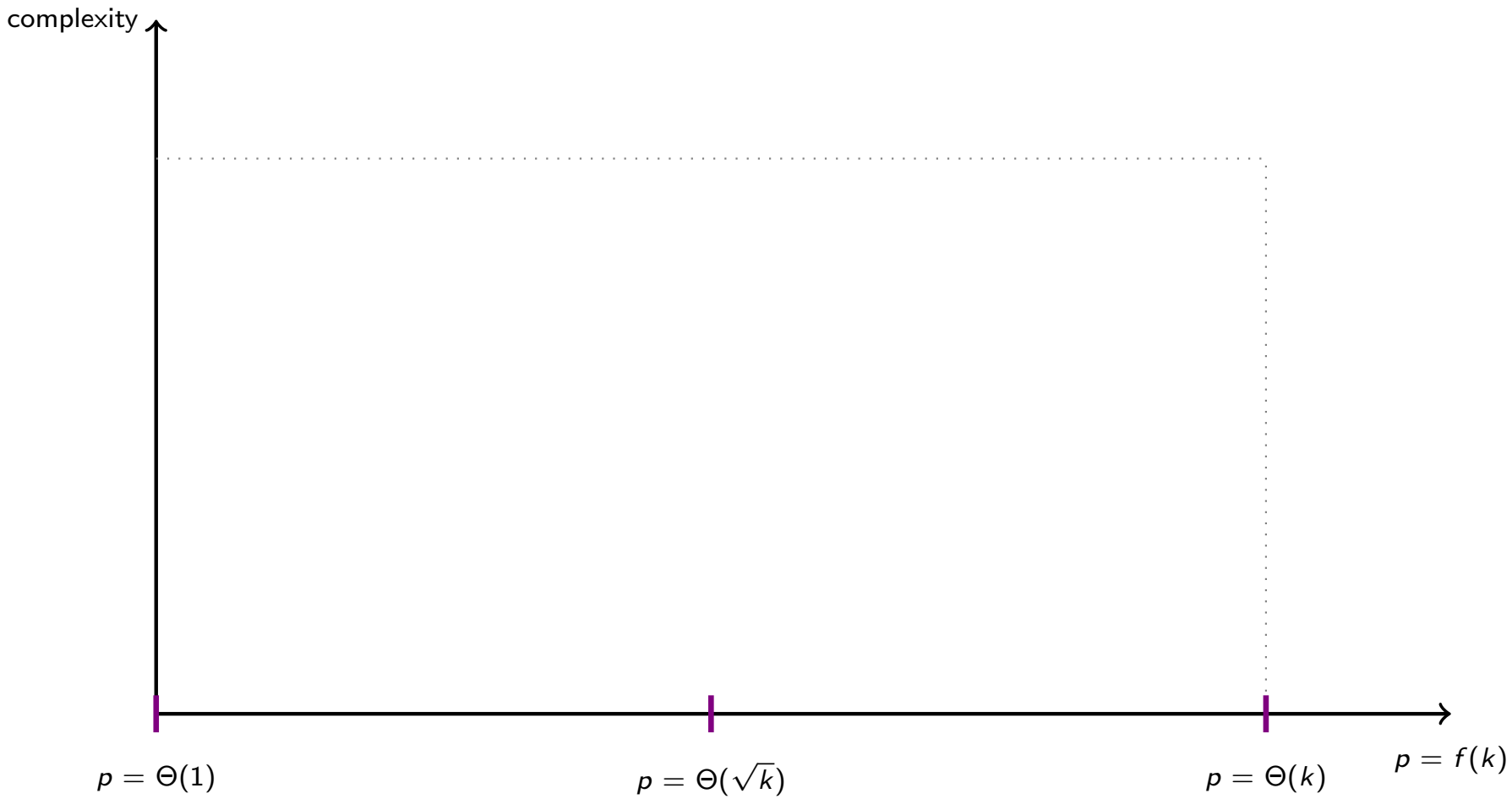
- p-Cluster Editing is at least as hard as $p_{\leqslant}$-Cluster Editing, but the converse is unclear.
- There is a polynomial-time reduction that ensures $p \leqslant 6k$ in both variants (technical).

Upper bound

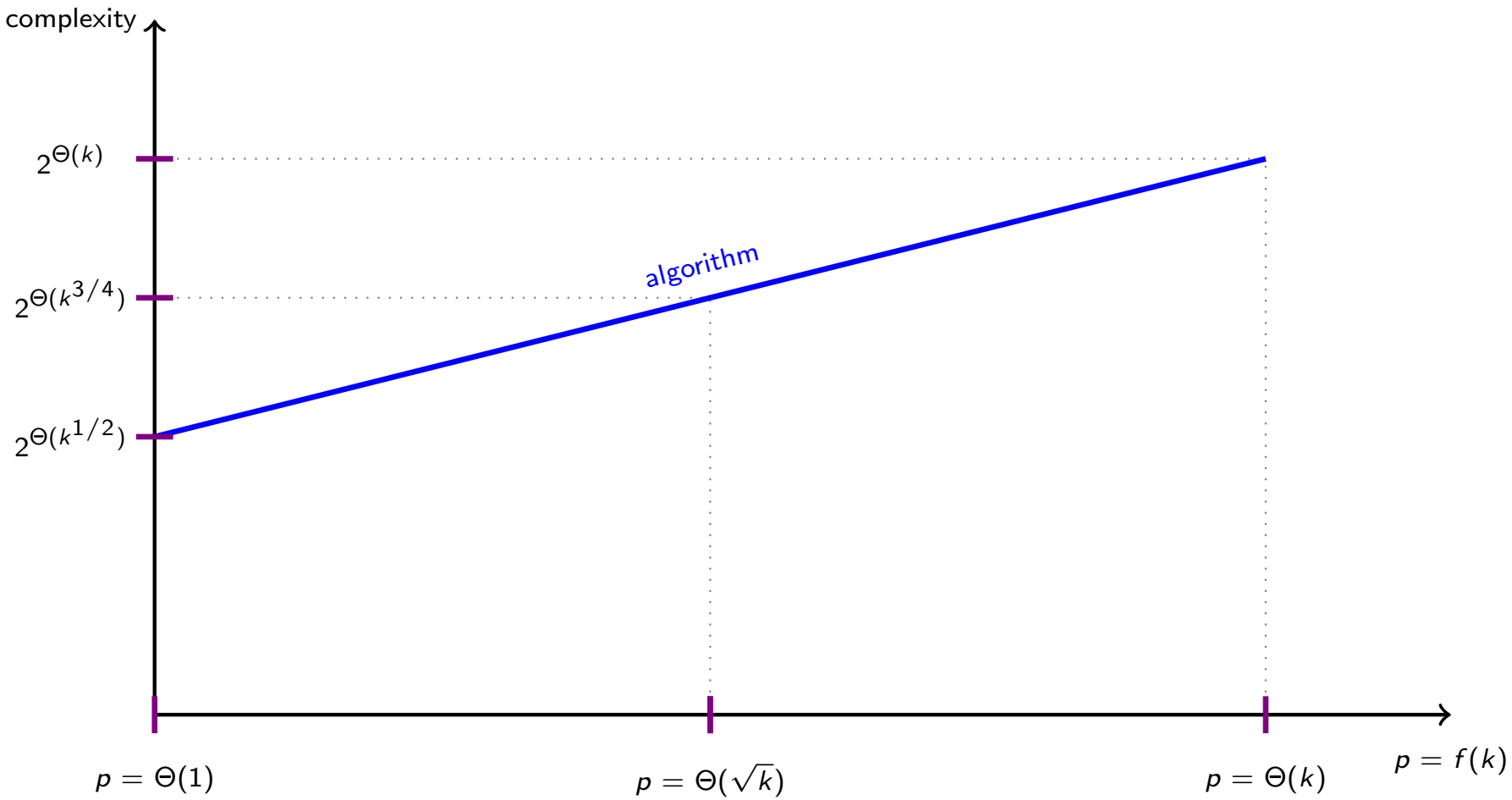
Theorem 1

p-Cluster Editing can be solved in $\mathcal{O}^\star(2^{\mathcal{O}(\sqrt{pk})})$ time.

Complexity



Complexity


complexity
$2^{\Theta(k)}$
$2^{\Theta(k^{3/4})}$
$2^{\Theta(k^{1/2})}$
algorithm
$p = \Theta(1)$
$p = \Theta(\sqrt{k})$
$p = \Theta(k)$
$p = f(k)$

Lower bound

Theorem 2

Lower bound

Theorem 2

A very technical statement of a reduction from 3-CNF-SAT

Lower bound

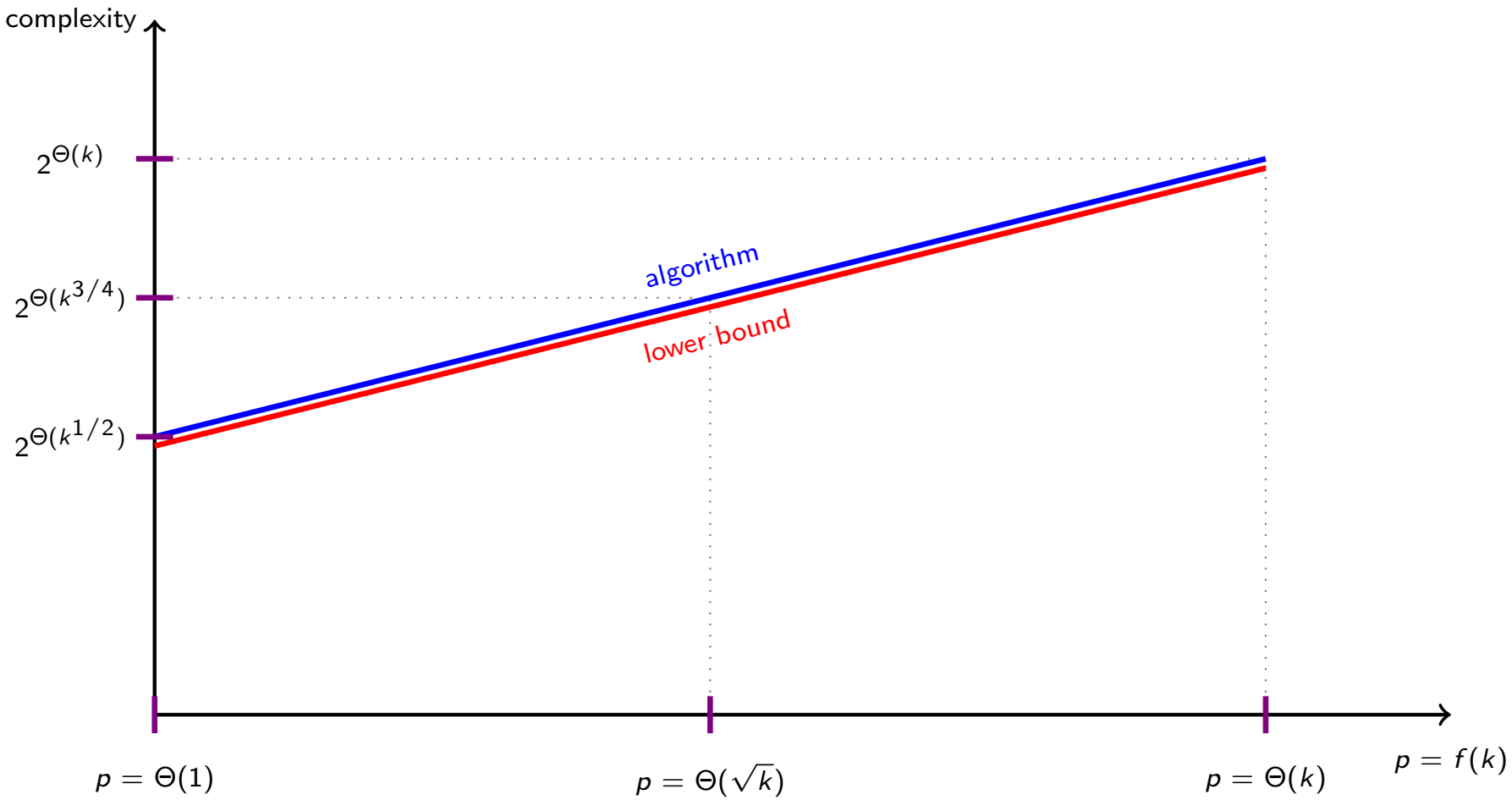
Theorem 2

A very technical statement of a reduction from 3-CNF-SAT

Corollary

Unless ETH fails, for every $\sigma \in [0,1]$ there exists $p(k) = \Theta(k^\sigma)$ such that there is no algorithm with running time $\mathcal{O}^\star(2^{o(\sqrt{pk})})$ solving $p_{\leqslant}$-Cluster Editing.

Complexity


complexity
$2^{\Theta(k)}$
$2^{\Theta(k^{3/4})}$
$2^{\Theta(k^{1/2})}$
algorithm
lower bound
$p = \Theta(1)$
$p = \Theta(\sqrt{k})$
$p = \Theta(k)$
$p = f(k)$

DP on subsets

- We start with showing a simple $\mathcal{O}^\star(3^n)$ DP on subsets.

DP on subsets

- We start with showing a simple $\mathcal{O}^\star(3^n)$ DP on subsets.
- **Set of states**: $2^{V(G)} \times \{0, 1, 2, \ldots, p\}$.

DP on subsets

- We start with showing a simple $\mathcal{O}^\star(3^n)$ DP on subsets.
- **Set of states**: $2^{V(G)} \times \{0, 1, 2, \ldots, p\}$.
- **Function**: $D[X][\ell]$ is the minimum number of editions needed to get ℓ clusters in $G[X]$.

DP on subsets

- We start with showing a simple $\mathcal{O}^\star(3^n)$ DP on subsets.
- **Set of states**: $2^{V(G)} \times \{0, 1, 2, \ldots, p\}$.
- **Function**: $D[X][\ell]$ is the minimum number of editions needed to get ℓ clusters in $G[X]$.
- **Recurrence**:

$$D[X][\ell] = \min_{\emptyset \neq Y \subseteq X} (D[X \setminus Y][\ell - 1] + |E(X \setminus Y, Y)| + |\overline{E}(Y, Y)|).$$

DP on subsets

- We start with showing a simple $\mathcal{O}^\star(3^n)$ DP on subsets.
- **Set of states**: $2^{V(G)} \times \{0, 1, 2, \ldots, p\}$.
- **Function**: $D[X][\ell]$ is the minimum number of editions needed to get ℓ clusters in $G[X]$.
- **Recurrence**:

$$D[X][\ell] = \min_{\emptyset \neq Y \subseteq X} (D[X \setminus Y][\ell - 1] + |E(X \setminus Y, Y)| + |\overline{E}(Y, Y)|).$$

- **Result**: $D[V(G)][p]$.

DP on subsets

- We start with showing a simple $\mathcal{O}^\star(3^n)$ DP on subsets.
- **Set of states**: $2^{V(G)} \times \{0, 1, 2, \ldots, p\}$.
- **Function**: $D[X][\ell]$ is the minimum number of editions needed to get ℓ clusters in $G[X]$.
- **Recurrence**:

$$D[X][\ell] = \min_{\emptyset \neq Y \subseteq X} (D[X \setminus Y][\ell - 1] + |E(X \setminus Y, Y)| + |\overline{E}(Y, Y)|).$$

- **Result**: $D[V(G)][p]$.
- **Time complexity**: for every subset we search through its subsets, so $\mathcal{O}^\star(3^n)$.

Trimming the space of states

- Which states do we really need if we know that in the solution there is at most k editions?

Trimming the space of states

- Which states do we really need if we know that in the solution there is at most k editions?
- Clearly, only subsets X such that $|E(X, V \setminus X)| \leqslant k$ are interesting.

Trimming the space of states

- Which states do we really need if we know that in the solution there is at most k editions?
- Clearly, only subsets X such that $|E(X, V \setminus X)| \leqslant k$ are interesting.

Definition

Partition (X, Y) of $V(G)$ is a *k-cut* if $|E(X, Y)| \leqslant k$.

Trimming the space of states

- Which states do we really need if we know that in the solution there is at most k editions?
- Clearly, only subsets X such that $|E(X, V \setminus X)| \leqslant k$ are interesting.

Definition

Partition (X, Y) of $V(G)$ is a *k-cut* if $|E(X, Y)| \leqslant k$.

- **Observation**: if someone gave us the set $\mathcal{N}$ of k-cuts, then we can perform the DP in $\mathcal{O}^\star(|\mathcal{N}|^2)$ time.

Strategy

- We show two lemmata:

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay **[standard]**

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay **[standard]**
  - The number of k-cuts in a YES-instance is at most $2^{\mathcal{O}(\sqrt{pk})}$.

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay **[standard]**
  - The number of k-cuts in a YES-instance is at most $2^{\mathcal{O}(\sqrt{pk})}$.
- **Algorithm**:

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay **[standard]**
  - The number of k-cuts in a YES-instance is at most $2^{\mathcal{O}(\sqrt{pk})}$.
- **Algorithm**:
  - we enumerate the k-cuts with polynomial delay;

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay **[standard]**
  - The number of k-cuts in a YES-instance is at most $2^{\mathcal{O}(\sqrt{pk})}$.
- **Algorithm**:
  - we enumerate the k-cuts with polynomial delay;
  - if we exceed the combinatorial bound, we say NO;

Strategy

- We show two lemmata:
  - k-cuts may be enumerated with polynomial delay **[standard]**
  - The number of k-cuts in a YES-instance is at most $2^{\mathcal{O}(\sqrt{pk})}$.
- **Algorithm**:
  - we enumerate the k-cuts with polynomial delay;
  - if we exceed the combinatorial bound, we say NO;
  - otherwise, we employ the DP on the set of k-cuts $\mathcal{N}$.

Bounding the number of k-cuts

- We aim to show that the number of k-cuts of a YES-instance is bounded by $2^{\mathcal{O}(\sqrt{pk})}$.

Bounding the number of k-cuts

- We aim to show that the number of k-cuts of a YES-instance is bounded by $2^{\mathcal{O}(\sqrt{pk})}$.
- **Trick**: all the k-cuts of a YES-instance are $2k$-cuts of some cluster graph with p clusters.

Bounding the number of k-cuts

- We aim to show that the number of k-cuts of a YES-instance is bounded by $2^{\mathcal{O}(\sqrt{pk})}$.
- **Trick**: all the k-cuts of a YES-instance are $2k$-cuts of some cluster graph with p clusters.
  - *Proof*: Having fixed some optimum solution, every k-cut can get at most k additional edges due to edge additions.

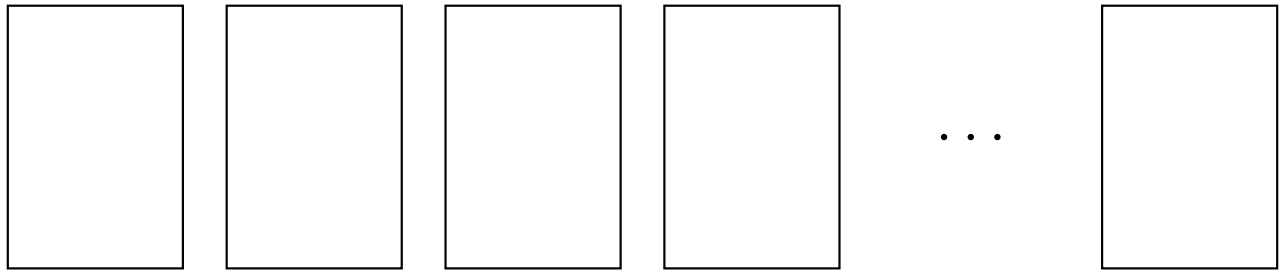
Bounding the number of k-cuts

- We aim to show that the number of k-cuts of a YES-instance is bounded by $2^{\mathcal{O}(\sqrt{pk})}$.
- **Trick**: all the k-cuts of a YES-instance are $2k$-cuts of some cluster graph with p clusters.
  - *Proof*: Having fixed some optimum solution, every k-cut can get at most k additional edges due to edge additions.
- Hence, we need to prove that the number of k-cuts of a cluster graph with p clusters is at most $2^{\mathcal{O}(\sqrt{pk})}$.

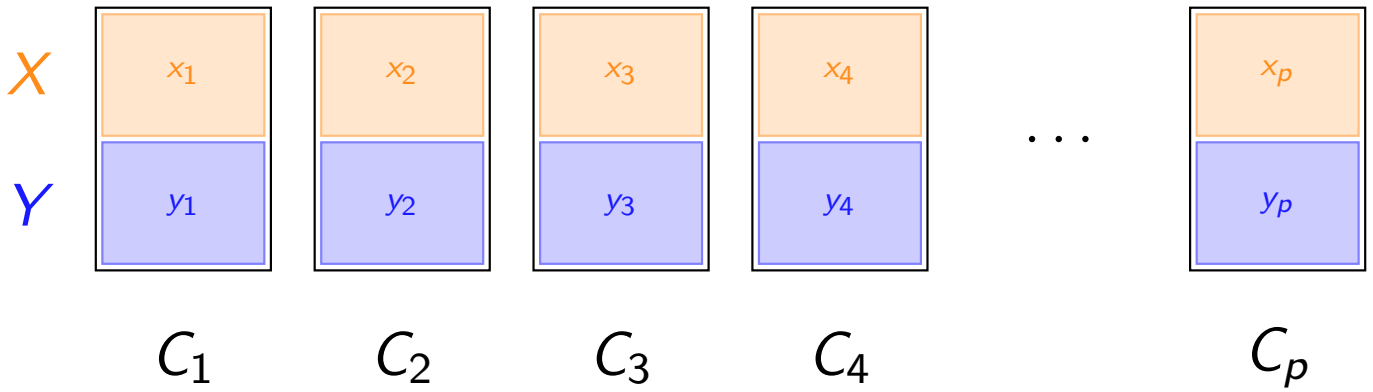
Bounding the number of k-cuts

- We aim to show that the number of k-cuts of a YES-instance is bounded by $2^{\mathcal{O}(\sqrt{pk})}$.
- **Trick**: all the k-cuts of a YES-instance are $2k$-cuts of some cluster graph with p clusters.
  - *Proof*: Having fixed some optimum solution, every k-cut can get at most k additional edges due to edge additions.
- Hence, we need to prove that the number of k-cuts of a cluster graph with p clusters is at most $2^{\mathcal{O}(\sqrt{pk})}$.
- **Note**: this is a purely combinatorial question!

k-cuts of a cluster graph

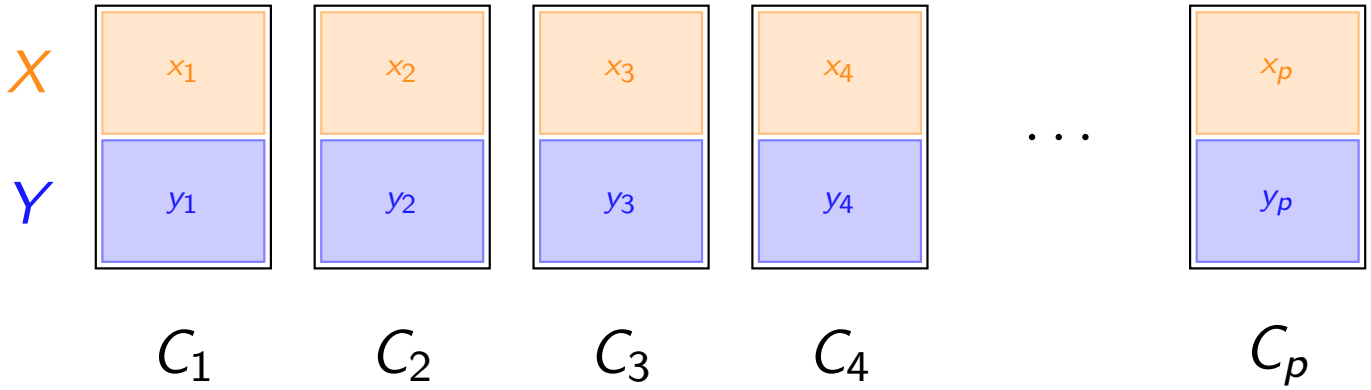


k-cuts of a cluster graph



- We first count the number of k-cuts (X, Y) such that $|C_i \cap X| = x_i$ and $|C_i \cap Y| = y_i$.

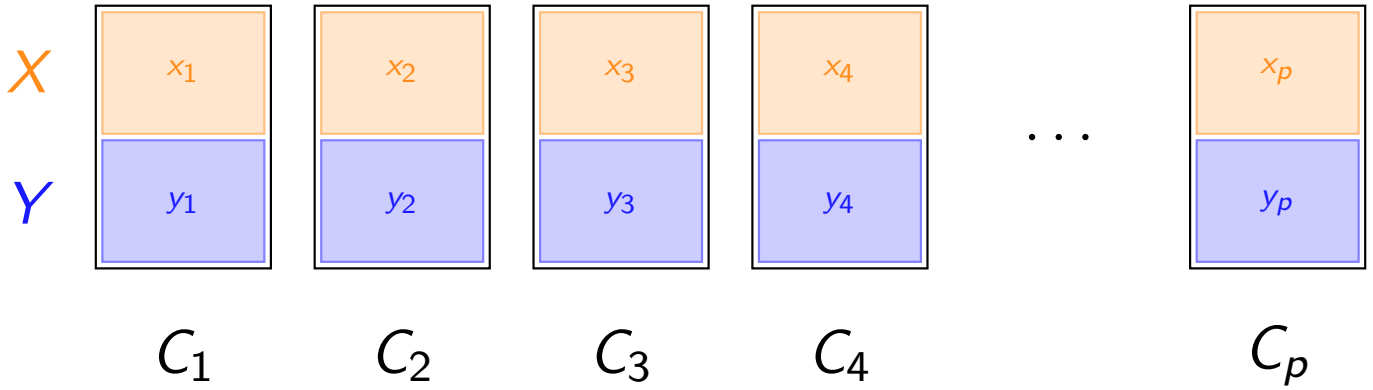
k-cuts of a cluster graph



- We first count the number of k-cuts (X, Y) such that $|C_i \cap X| = x_i$ and $|C_i \cap Y| = y_i$.
- This number is equal to

$$\binom{x_1 + y_1}{x_1} \cdot \binom{x_2 + y_2}{x_2} \cdot \ldots \cdot \binom{x_p + y_p}{x_p}$$

k-cuts of a cluster graph



- We first count the number of k-cuts (X, Y) such that $|C_i \cap X| = x_i$ and $|C_i \cap Y| = y_i$.
- This number is equal to

$$\binom{x_1 + y_1}{x_1} \cdot \binom{x_2 + y_2}{x_2} \cdot \ldots \cdot \binom{x_p + y_p}{x_p}$$

- **Note**: $\sum_{i=1}^{p} x_i y_i \leqslant k$.

k-cuts of a cluster graph

- **Inequality**: for nonnegative integers a, b, $\binom{a+b}{a} \leqslant 4^{\sqrt{ab}}$.

k-cuts of a cluster graph

- **Inequality**: for nonnegative integers a, b, $\binom{a+b}{a} \leqslant 4^{\sqrt{ab}}$.
  - *Proof*: Use $\binom{a+b}{b} \leqslant \frac{(a+b)^{a+b}}{a^a b^b}$ and then differentiate three times.

k-cuts of a cluster graph

- **Inequality**: for nonnegative integers a, b, $\binom{a+b}{a} \leqslant 4^{\sqrt{ab}}$.
  - *Proof*: Use $\binom{a+b}{b} \leqslant \frac{(a+b)^{a+b}}{a^a b^b}$ and then differentiate three times.
- Hence,

$$\prod_{i=1}^{p} \binom{x_i + y_i}{x_i} \leqslant \prod_{i=1}^{p} 4^{\sqrt{x_i y_i}} \leqslant 4^{\sqrt{pk}},$$

k-cuts of a cluster graph

- **Inequality**: for nonnegative integers a, b, $\binom{a+b}{a} \leqslant 4^{\sqrt{ab}}$.
  - *Proof*: Use $\binom{a+b}{b} \leqslant \frac{(a+b)^{a+b}}{a^a b^b}$ and then differentiate three times.
- Hence,

$$\prod_{i=1}^{p} \binom{x_i + y_i}{x_i} \leqslant \prod_{i=1}^{p} 4^{\sqrt{x_i y_i}} \leqslant 4^{\sqrt{pk}},$$

- because of Cauchy-Schwartz inequality:

$$\sqrt{kp} \geqslant \sqrt{\sum_{i=1}^{p} (\sqrt{x_i y_i})^2} \cdot \sqrt{\sum_{i=1}^{p} 1^2} \geqslant \sum_{i=1}^{p} \sqrt{x_i y_i} \cdot 1.$$

k-cuts of a cluster graph

- **Inequality**: for nonnegative integers a, b, $\binom{a+b}{a} \leqslant 4^{\sqrt{ab}}$.
  - *Proof*: Use $\binom{a+b}{b} \leqslant \frac{(a+b)^{a+b}}{a^a b^b}$ and then differentiate three times.
- Hence,

$$\prod_{i=1}^{p} \binom{x_i + y_i}{x_i} \leqslant \prod_{i=1}^{p} 4^{\sqrt{x_i y_i}} \leqslant 4^{\sqrt{pk}},$$

- because of Cauchy-Schwartz inequality:

$$\sqrt{kp} \geqslant \sqrt{\sum_{i=1}^{p} (\sqrt{x_i y_i})^2} \cdot \sqrt{\sum_{i=1}^{p} 1^2} \geqslant \sum_{i=1}^{p} \sqrt{x_i y_i} \cdot 1.$$

- The number of choices (x_i, y_i) is also bounded by $2^{\mathcal{O}(\sqrt{pk})}$ (easy, but we need $p = \mathcal{O}(k)$).

Lower bound

- The lower bound is a technical reduction from 3-CNF-SAT.

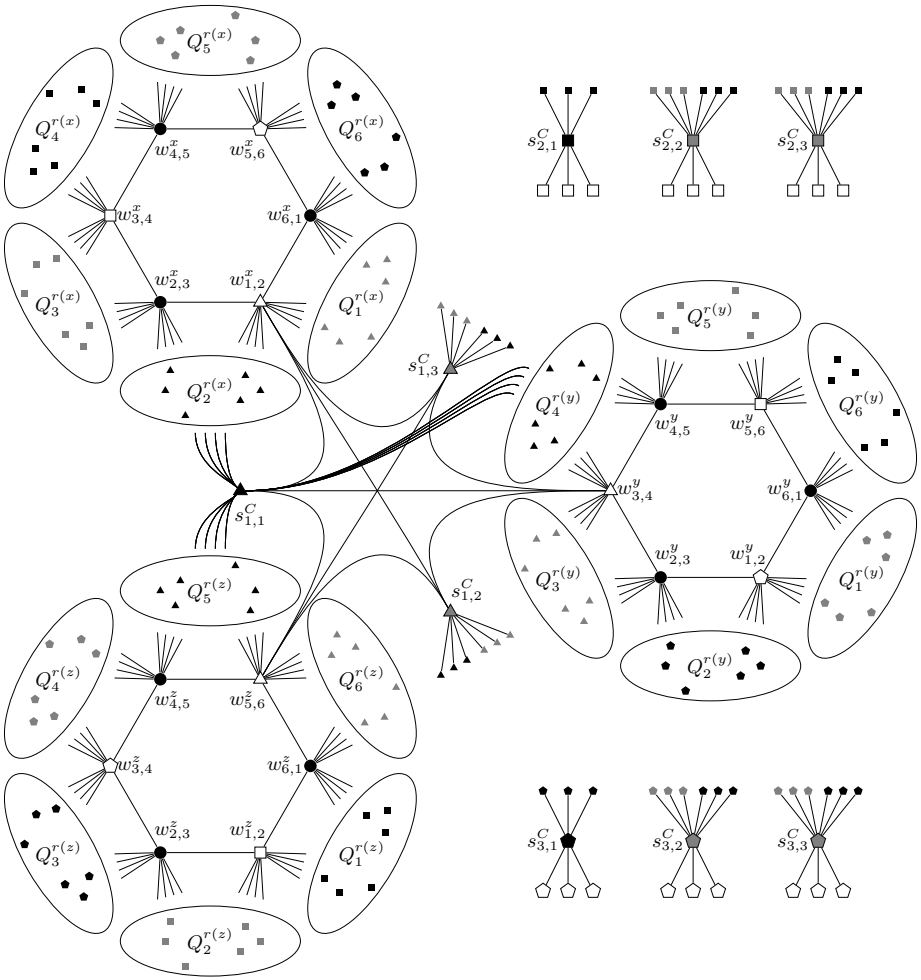
Lower bound

- The lower bound is a technical reduction from 3-CNF-SAT.
- Involves preprocessing of the formula, variable grouping, complicated gadgets, and needs very careful analysis.

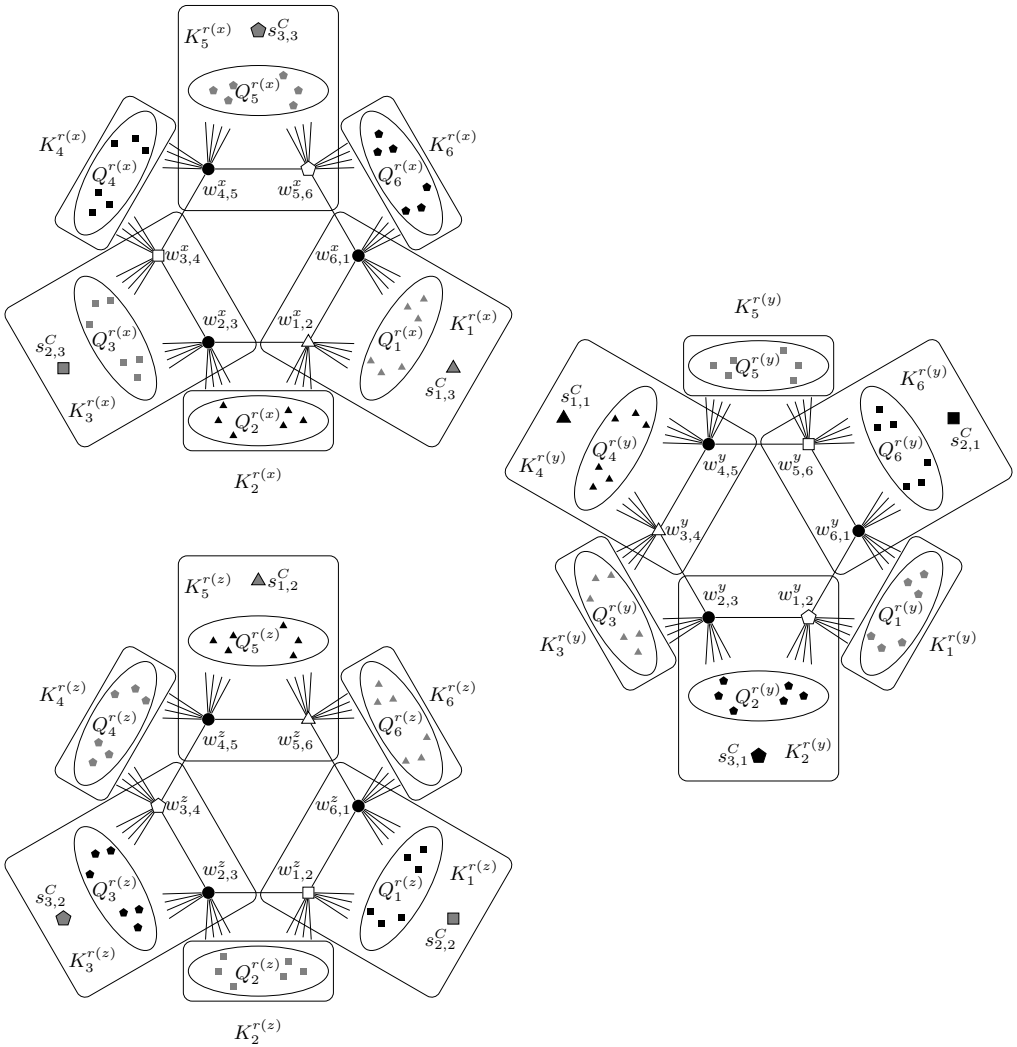
Lower bound

- The lower bound is a technical reduction from 3-CNF-SAT.
- Involves preprocessing of the formula, variable grouping, complicated gadgets, and needs very careful analysis.
- Hence, we will present just some pictures of gadgets.

Gadgets


$Q_5^{r(x)}$
$Q_4^{r(x)}$
$Q_6^{r(x)}$
$w_{4,5}^x$
$w_{5,6}^x$
$w_{3,4}^x$
$w_{6,1}^x$
$w_{2,3}^x$
$w_{1,2}^x$
$Q_3^{r(x)}$
$Q_1^{r(x)}$
$Q_2^{r(x)}$
$s_{2,1}^C$
$s_{2,2}^C$
$s_{2,3}^C$
$s_{1,3}^C$
$s_{1,1}^C$
$s_{1,2}^C$
$Q_5^{r(y)}$
$Q_4^{r(y)}$
$Q_6^{r(y)}$
$w_{4,5}^y$
$w_{5,6}^y$
$w_{3,4}^y$
$w_{6,1}^y$
$w_{2,3}^y$
$w_{1,2}^y$
$Q_3^{r(y)}$
$Q_1^{r(y)}$
$Q_2^{r(y)}$
$Q_5^{r(z)}$
$Q_4^{r(z)}$
$Q_6^{r(z)}$
$w_{4,5}^z$
$w_{5,6}^z$
$w_{3,4}^z$
$w_{6,1}^z$
$w_{2,3}^z$
$w_{1,2}^z$
$Q_3^{r(z)}$
$Q_1^{r(z)}$
$Q_2^{r(z)}$
$s_{3,1}^C$
$s_{3,2}^C$
$s_{3,3}^C$

Gadgets


$K_5^{r(x)}$
$s_{3,3}^C$
$Q_5^{r(x)}$
$K_4^{r(x)}$
$Q_4^{r(x)}$
$K_6^{r(x)}$
$Q_6^{r(x)}$
$w_{4,5}^x$
$w_{5,6}^x$
$w_{3,4}^x$
$w_{6,1}^x$
$w_{2,3}^x$
$w_{1,2}^x$
$s_{2,3}^C$
$Q_3^{r(x)}$
$K_1^{r(x)}$
$Q_1^{r(x)}$
$s_{1,3}^C$
$K_3^{r(x)}$
$Q_2^{r(x)}$
$K_2^{r(x)}$
$K_5^{r(y)}$
$s_{1,1}^C$
$Q_5^{r(y)}$
$K_6^{r(y)}$
$K_4^{r(y)}$
$Q_4^{r(y)}$
$s_{2,1}^C$
$Q_6^{r(y)}$
$w_{4,5}^y$
$w_{5,6}^y$
$w_{3,4}^y$
$w_{6,1}^y$
$w_{2,3}^y$
$w_{1,2}^y$
$Q_3^{r(y)}$
$Q_1^{r(y)}$
$K_3^{r(y)}$
$K_1^{r(y)}$
$Q_2^{r(y)}$
$s_{3,1}^C$
$K_2^{r(y)}$
$K_5^{r(z)}$
$s_{1,2}^C$
$Q_5^{r(z)}$
$K_4^{r(z)}$
$Q_4^{r(z)}$
$K_6^{r(z)}$
$Q_6^{r(z)}$
$w_{4,5}^z$
$w_{5,6}^z$
$w_{3,4}^z$
$w_{6,1}^z$
$w_{2,3}^z$
$w_{1,2}^z$
$s_{3,2}^C$
$Q_3^{r(z)}$
$K_1^{r(z)}$
$Q_1^{r(z)}$
$s_{2,2}^C$
$K_3^{r(z)}$
$Q_2^{r(z)}$
$K_2^{r(z)}$

Open problems

- More problems solvable via enumeration of k-cuts?

Open problems

- More problems solvable via enumeration of k-cuts?
- Our reduction gives $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bound for constant p, but only for $p \geqslant 6$.

Open problems

- More problems solvable via enumeration of k-cuts?
- Our reduction gives $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bound for constant p, but only for $p \geqslant 6$.
  - **Open**: What about $p < 6$?

Open problems

- More problems solvable via enumeration of k-cuts?
- Our reduction gives $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bound for constant p, but only for $p \geqslant 6$.
  - **Open**: What about $p < 6$?
- Links between parameterized complexity and approximation:

Open problems

- More problems solvable via enumeration of k-cuts?
- Our reduction gives $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bound for constant p, but only for $p \geqslant 6$.
  - **Open**: What about $p < 6$?
- Links between parameterized complexity and approximation:
  - **Open**: Is there a PTAS for $p = \sqrt{k}$?

Open problems

- More problems solvable via enumeration of k-cuts?
- Our reduction gives $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bound for constant p, but only for $p \geqslant 6$.
  - **Open**: What about $p < 6$?
- Links between parameterized complexity and approximation:
  - **Open**: Is there a PTAS for $p = \sqrt{k}$?
  - **Open**: Can we get EPTAS instead of PTAS?

Open problems

- More problems solvable via enumeration of k-cuts?
- Our reduction gives $\mathcal{O}^\star(2^{o(\sqrt{k})})$ lower bound for constant p, but only for $p \geqslant 6$.
  - **Open**: What about $p < 6$?
- Links between parameterized complexity and approximation:
  - **Open**: Is there a PTAS for $p = \sqrt{k}$?
  - **Open**: Can we get EPTAS instead of PTAS?
- **Thanks for attention!**