Optimal data reduction for graph coloring using low-degree polynomials

Astrid Pieterse

Networks Training Week

October 31, 2017

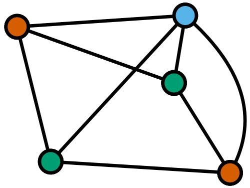
Technische Universiteit
Eindhoven
University of Technology

The q-Coloring problem

Can the vertices of a graph be colored with at most q colors?

- Focus on $q = 3$
- red, green, blue

NP-hard, use a parameterized approach



Which parameter(s)?

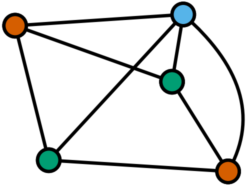
- Number of colors
  - Uninteresting
- Structural parameters
  - In this talk: Vertex Cover

The q-Coloring problem

Can the vertices of a graph be colored with at most q colors?

- Focus on $q = 3$
- red, green, blue

NP-hard, use a parameterized approach



Which parameter(s)?

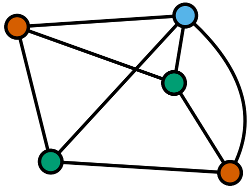
- Number of colors
  - Uninteresting
- Structural parameters
  - In this talk: Vertex Cover

The q-Coloring problem

Can the vertices of a graph be colored with at most q colors?

- Focus on $q = 3$
- red, green, blue

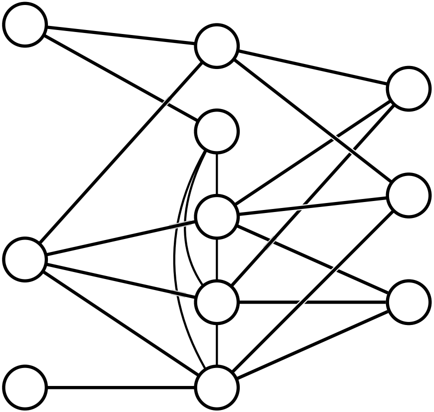
NP-hard, use a parameterized approach



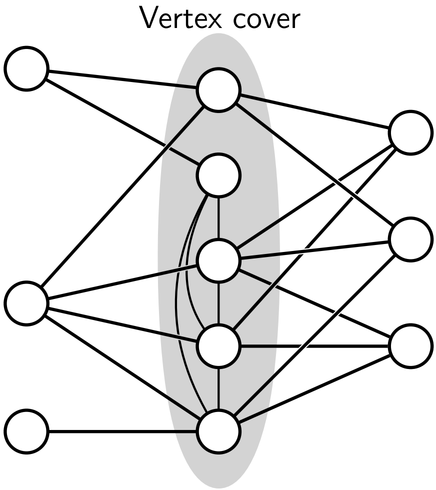
Which parameter(s)?

- Number of colors
  - Uninteresting
- Structural parameters
  - In this talk: Vertex Cover

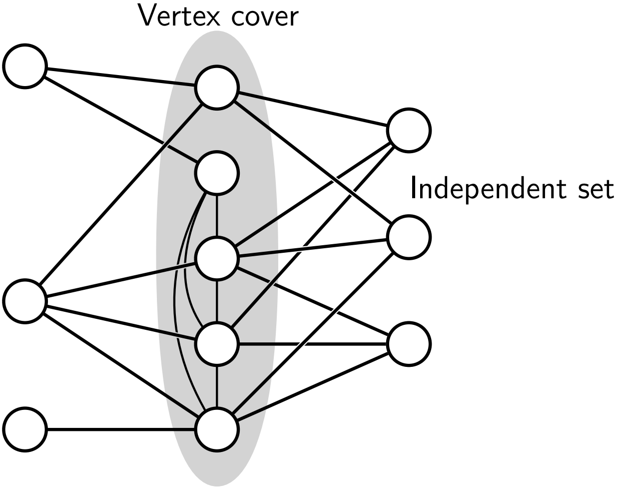
Vertex Cover



Vertex Cover



Vertex Cover



Kernelization

Efficiently reduce the size of an input instance

- Resulting size depends on parameter value
- Provably small
- Provably correct

Kernel for 3-Coloring

Polynomial-time algorithm that, given instance (G, k), outputs (G', k') such that

- $|G'|$ and k' are bounded by $f(k)$
- G is 3-Colorable if and only if G' is 3-Colorable

Kernel for 3-Coloring

Previous work

Jansen and Kratsch [Inf Comput. 2013] showed that

- 3-Coloring parameterized by VC has a kernel of bitsize $O(k^3)$

Theorem

3-*Coloring parameterized by Vertex Cover has a kernel with* $O(k^2)$ *vertices and bitsize* $O(k^2 \log k)$.

Matching lower bound

Kernel for 3-Coloring

Previous work

Jansen and Kratsch [Inf Comput. 2013] showed that

- 3-Coloring parameterized by VC has a kernel of bitsize $O(k^3)$

Theorem

3-*Coloring parameterized by Vertex Cover has a kernel with* $O(k^2)$ *vertices and bitsize* $O(k^2 \log k)$.

Matching lower bound

Kernel for 3-Coloring

Previous work

Jansen and Kratsch [Inf Comput. 2013] showed that

- 3-Coloring parameterized by VC has a kernel of bitsize $O(k^3)$

Theorem

3-*Coloring parameterized by Vertex Cover has a kernel with* $O(k^2)$ *vertices and bitsize* $O(k^2 \log k)$.

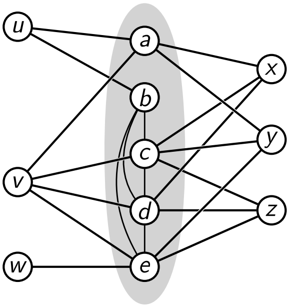
Matching lower bound

Kernel for 3-Coloring

Kernel: general idea

We have graph G, with vertex cover VC
The remaining vertices form independent set IS

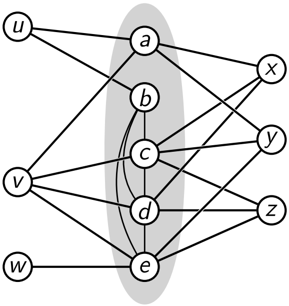
- IS may be large compared to VC
- Find redundant vertices in IS
  - Any coloring of $G - u$ can be extended to G
  - Example: u and w
- Similarly, find redundant edges



Kernel: general idea

We have graph G, with vertex cover VC
The remaining vertices form independent set IS

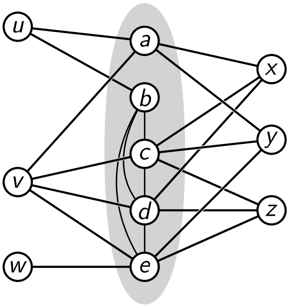
- IS may be large compared to VC
- Find redundant vertices in IS
  - Any coloring of $G - u$ can be extended to G
  - Example: u and w
- Similarly, find redundant edges



Kernel: general idea

We have graph G, with vertex cover VC
The remaining vertices form independent set IS

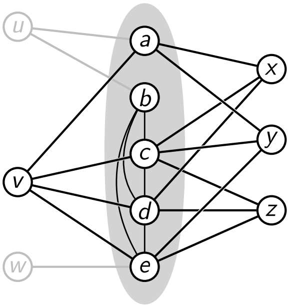
- IS may be large compared to VC
- Find redundant vertices in IS
  - Any coloring of $G - u$ can be extended to G
  - Example: u and w
- Similarly, find redundant edges



Kernel: general idea

We have graph G, with vertex cover VC
The remaining vertices form independent set IS

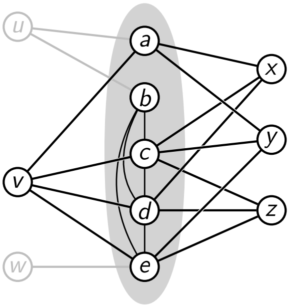
- IS may be large compared to VC
- Find redundant vertices in IS
  - Any coloring of $G - u$ can be extended to G
  - Example: u and w
- Similarly, find redundant edges



Kernel: general idea

We have graph G, with vertex cover VC
The remaining vertices form independent set IS

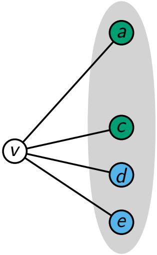
- IS may be large compared to VC
- Find redundant vertices in IS
  - Any coloring of $G - u$ can be extended to G
  - Example: u and w
- Similarly, find redundant edges



Kernel: general idea

Vertices in IS can be colored independently

- Each vertex in IS corresponds to a constraint
  - Neighborhood does not use all 3 colors
- Gives constraints on the coloring of VC
  - If some coloring of VC satisfies all constraints, it can be extended to IS



Alternatively, if for all $S \subseteq N(v)$ with $|S| = 3$

Some color is used twice for S

$$\Updownarrow$$

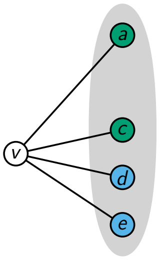
coloring can be extended to v

- Proof by contradiction

Kernel: general idea

Vertices in IS can be colored independently

- Each vertex in IS corresponds to a constraint
  - Neighborhood does not use all 3 colors
- Gives constraints on the coloring of VC
  - If some coloring of VC satisfies all constraints, it can be extended to IS



Alternatively, if for all $S \subseteq N(v)$ with $|S| = 3$

Some color is used twice for S

$$\Updownarrow$$

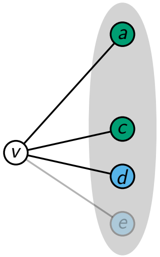
coloring can be extended to v

- Proof by contradiction

Kernel: general idea

Vertices in IS can be colored independently

- Each vertex in IS corresponds to a constraint
  - Neighborhood does not use all 3 colors
- Gives constraints on the coloring of VC
  - If some coloring of VC satisfies all constraints, it can be extended to IS



Alternatively, if for all $S \subseteq N(v)$ with $|S| = 3$

Some color is used twice for S

$$\Updownarrow$$

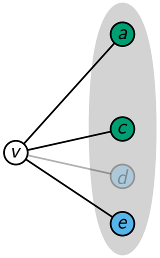
coloring can be extended to v

- Proof by contradiction

Kernel: general idea

Vertices in IS can be colored independently

- Each vertex in IS corresponds to a constraint
  - Neighborhood does not use all 3 colors
- Gives constraints on the coloring of VC
  - If some coloring of VC satisfies all constraints, it can be extended to IS



Alternatively, if for all $S \subseteq N(v)$ with $|S| = 3$

Some color is used twice for S

$\Updownarrow$

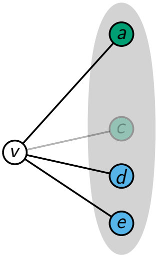
coloring can be extended to v

- Proof by contradiction

Kernel: general idea

Vertices in IS can be colored independently

- Each vertex in IS corresponds to a constraint
  - Neighborhood does not use all 3 colors
- Gives constraints on the coloring of VC
  - If some coloring of VC satisfies all constraints, it can be extended to IS



Alternatively, if for all $S \subseteq N(v)$ with $|S| = 3$

Some color is used twice for S

$$\Updownarrow$$

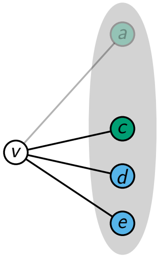
coloring can be extended to v

- Proof by contradiction

Kernel: general idea

Vertices in IS can be colored independently

- Each vertex in IS corresponds to a constraint
  - Neighborhood does not use all 3 colors
- Gives constraints on the coloring of VC
  - If some coloring of VC satisfies all constraints, it can be extended to IS



Alternatively, if for all $S \subseteq N(v)$ with $|S| = 3$

Some color is used twice for S

$$\Updownarrow$$

coloring can be extended to v

- Proof by contradiction

Finding redundant constraints

Finding redundant constraints

Let L be a set of polynomial equalities of degree at most d, over n boolean variables.

$$
\begin{aligned}
L = \{ & \\
& x_1x_2 + x_2x_3 + x_4 \equiv_2 0 \\
& x_1 + 1 \equiv_2 0 \\
& x_3x_1 + 1 \equiv_2 0 \\
& \dots \\
\} &
\end{aligned}
$$

Theorem [Jansen and P. MFCS 2016]

There is a polynomial-time algorithm that outputs $L' \subseteq L$, s.t.

- *An assignment satisfies L if and only if it satisfies L' and*
- $|L'| \leq n^d + 1$

Example

Let $x, y, z \in \{0, 1\}$ (where 0 is false, 1 is true), let

$$L = \{$$
$$x + y = 1$$
$$z - y = 0$$
$$x + z = 1$$
$$\}$$

- The last constraint is redundant

Example

Let $x, y, z \in \{0, 1\}$ (where 0 is false, 1 is true), let

$$L = \{$$

$$\frac{\begin{array}{c} x + y = 1 \\ z - y = 0 \end{array}}{x + z = 1} +$$

$$\}$$

- The last constraint is redundant

Example

Let $x, y, z \in \{0, 1\}$ (where 0 is false, 1 is true), let

$$L = \{$$

$$\frac{\begin{array}{c} x + y = 1 \\ z - y = 0 \end{array}}{x + z = 1} +$$

$$\}$$

- The last constraint is redundant

Example

Let $x, y, z \in \{0, 1\}$ (where 0 is false, 1 is true), let

$$L = \{ \quad \begin{array}{r} x + y = 1 \\ z - y = 0 \\ \hline x + z = 1 \end{array} + \quad \}$$

$$L' = \{ \quad \begin{array}{r} x + y = 1 \\ z - y = 0 \end{array} \quad \}$$

- The last constraint is redundant

Finding redundant constraints

$$L := \{y + 1 \equiv_2 0, x + y + xy \equiv_2 0, x \equiv_2 0\}$$

- Represent the equalities in a matrix:

$$\begin{array}{cc} & \begin{array}{cccc} x & y & xy & 1 \end{array} \\ \begin{array}{c} f \\ g \\ \vdots \\ h \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

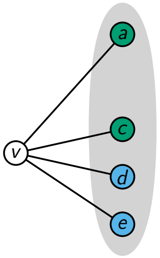
- Apply gaussian elimination to find a basis of the row-space
  - Size bounded by #columns!

Kernel for 3-Coloring

Modeling vertices as constraints

Polynomial equalities

- Create 3 boolean variables for each vertex in VC.
  - a a a indicate the color of a
- For each vertex v in IS, $S \subseteq N(v)$ with $|S| = 3$
  - Constraint: S does not use all 3 colors.



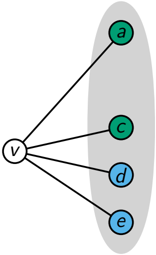
Which polynomial to use?

- Needs to have degree ≤ 2

Modeling vertices as constraints

Polynomial equalities

- Create 3 boolean variables for each vertex in VC.
  - a a a indicate the color of a
- For each vertex v in IS, $S \subseteq N(v)$ with $|S| = 3$
  - Constraint: S does not use all 3 colors.



Which polynomial to use?

- Needs to have degree ≤ 2

Polynomial equalities for 3-Coloring

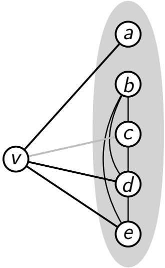
- 3 variables for each vertex in $\textsc{vc}$ a a a

Let $v \in \textsc{is}$, for each $S \subseteq N(v) : |S| = 3$

- Polynomial equality of degree 2
  - For $S = \{a, d, e\}$:

$$\begin{aligned} &a \wedge d + a \wedge e + d \wedge e + \\ &a \wedge d + a \wedge e + d \wedge e + \\ &a \wedge d + a \wedge e + d \wedge e \equiv_2 1 \end{aligned}$$

- Expresses: a,d, and e do not use all 3 colors
  - Three equal colors gives $3 \equiv_2 1$
  - Two equal colors gives 1
  - Three different colors gives 0



Polynomial equalities for 3-Coloring

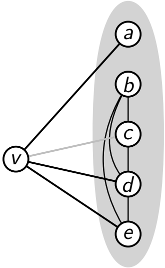
- 3 variables for each vertex in VC a a a

Let $v \in \text{IS}$, for each $S \subseteq N(v) : |S| = 3$

- Polynomial equality of degree 2
  - For $S = \{a, d, e\}$:

$$a \wedge d + a \wedge e + d \wedge e +$$
$$a \wedge d + a \wedge e + d \wedge e +$$
$$a \wedge d + a \wedge e + d \wedge e \equiv_2 1$$

- Expresses: a,d, and e do not use all 3 colors
  - Three equal colors gives $3 \equiv_2 1$
  - Two equal colors gives 1
  - Three different colors gives 0



Polynomial equalities for 3-Coloring

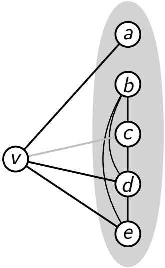
- 3 variables for each vertex in VC a a a

Let $v \in$ IS, for each $S \subseteq N(v) : |S| = 3$

- Polynomial equality of degree 2
  - For $S = \{a, d, e\}$:

$$a \wedge d + a \wedge e + d \wedge e +$$
$$a \wedge d + a \wedge e + d \wedge e +$$
$$a \wedge d + a \wedge e + d \wedge e \equiv_2 1$$

- Expresses: a,d, and e do not use all 3 colors
  - Three equal colors gives $3 \equiv_2 1$
  - Two equal colors gives 1
  - Three different colors gives 0



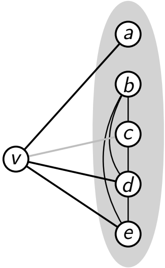
Polynomial equalities for 3-Coloring

- 3 variables for each vertex in VC a a a

Let $v \in \text{IS}$, for each $S \subseteq N(v) : |S| = 3$

- Polynomial equality of degree 2
  - For $S = \{a, d, e\}$:

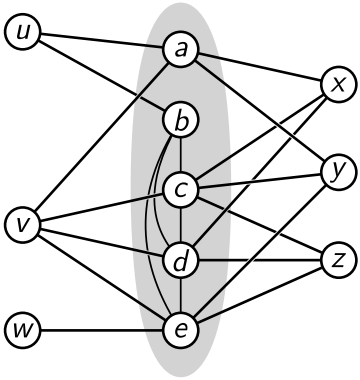
$$
\begin{aligned}
&a \wedge d + a \wedge e + d \wedge e + \\
&a \wedge d + a \wedge e + d \wedge e + \\
&a \wedge d + a \wedge e + d \wedge e \equiv_2 1
\end{aligned}
$$



- Expresses: *a*,*d*, and *e* do not use all 3 colors
  - Three equal colors gives $3 \equiv_2 1$
  - Two equal colors gives 1
  - Three different colors gives 0

Kernel for 3-Coloring

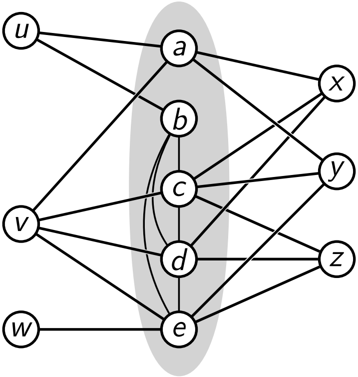
- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
$C(a, c, d)$
$C(a, d, e)$
$C(c, d, e)$
$C(a, c, e)$
x:
$C(a, c, d)$
y:
$C(a, c, e)$
z:
$C(c, d, e)$

Kernel for 3-Coloring

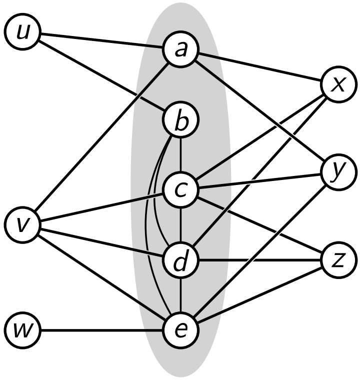
- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
$C(a, c, d)$
$C(a, d, e)$
$C(c, d, e)$
$C(a, c, e)$
x:
$C(a, c, d)$
y:
$C(a, c, e)$
z:
$C(c, d, e)$

Kernel for 3-Coloring

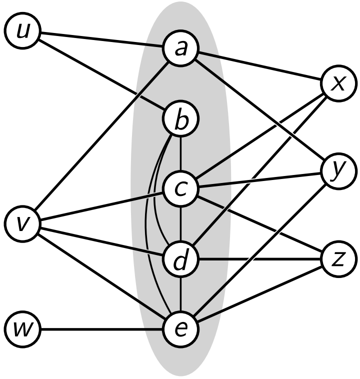
- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
$\cancel{C(a,c,d)}$
$C(a,d,e)$
$\cancel{C(c,d,e)}$
$\cancel{C(a,c,e)}$
x:
$C(a,c,d)$
y:
$C(a,c,e)$
z:
$\cancel{C(c,d,e)}$

Kernel for 3-Coloring

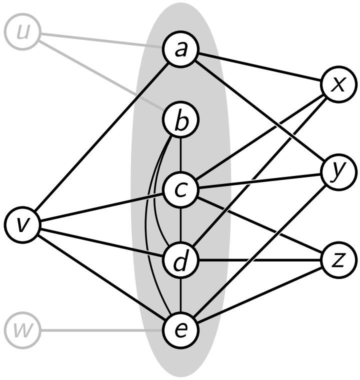
- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
$\cancel{C(a, c, d)}$
$C(a, d, e)$
$\cancel{C(c, d, e)}$
$\cancel{C(a, c, e)}$
x:
$C(a, c, d)$
y:
$C(a, c, e)$
z:
$\cancel{C(c, d, e)}$

Kernel for 3-Coloring

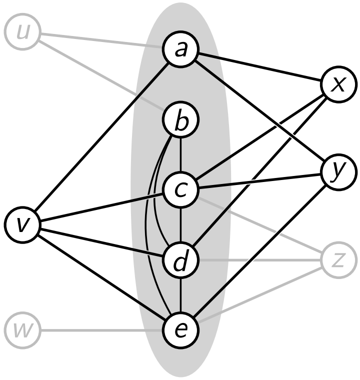
- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
$\cancel{C(a, c, d)}$
$C(a, d, e)$
$\cancel{C(c, d, e)}$
$\cancel{C(a, c, e)}$
x:
$C(a, c, d)$
y:
$C(a, c, e)$
z:
$\cancel{C(c, d, e)}$

Kernel for 3-Coloring

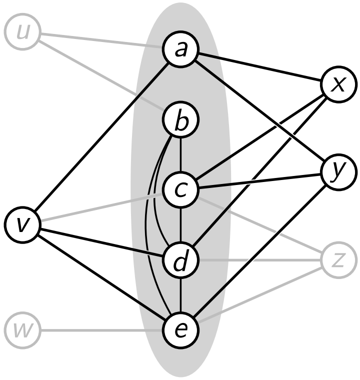
- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
$\cancel{C(a,c,d)}$
$C(a,d,e)$
$\cancel{C(c,d,e)}$
$\cancel{C(a,c,e)}$
x:
$C(a,c,d)$
y:
$C(a,c,e)$
z:
$\cancel{C(c,d,e)}$

Kernel for 3-Coloring

- Model vertices in IS by constraints
- Use Theorem to find subset L' of relevant constraints
- Keep only vertices and edges used for relevant constraints



u, w: $\emptyset$
v:
~~$C(a, c, d)$~~
$C(a, d, e)$
~~$C(c, d, e)$~~
~~$C(a, c, e)$~~
x:
$C(a, c, d)$
y:
$C(a, c, e)$
z:
~~$C(c, d, e)$~~

Kernel size

- Number of constraints $O(\#vars^{degree})$
  - $3 \cdot k$ variables
  - degree 2
- Number of constraints $O((3k)^2)$
- Constraint corresponds to ≤ 1 vertex and ≤ 3 edges
- Encode graph in $O(k^2 \log k)$ bits

Theorem

3-*Coloring parameterized by Vertex Cover has a kernel with $O(k^2)$ vertices and bitsize $O(k^2 \log k)$.*

q-Coloring

Coloring with q colors

Theorem

q-Coloring parameterized by Vertex Cover has a kernel with $O(k^{q-1})$ vertices and bitsize $O(k^{q-1} \log k)$.

Can we do better?

Theorem

q-Coloring parameterized by Vertex Cover has no kernel of bitsize $O(k^{q-1-\varepsilon})$, unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

q-Coloring

Coloring with q colors

Theorem

q-Coloring parameterized by Vertex Cover has a kernel with $O(k^{q-1})$ vertices and bitsize $O(k^{q-1} \log k)$.

Can we do better?

Theorem

q-Coloring parameterized by Vertex Cover has no kernel of bitsize $O(k^{q-1-\varepsilon})$, unless $\mathsf{NP} \subseteq \mathsf{coNP}/\mathsf{poly}$.

Conclusion

q-Coloring has

- a kernel with bitsize $O(k^{q-1} \log k)$
- no kernel with bitsize $O(k^{q-1-\varepsilon})$ unless NP $\subseteq$ coNP/poly

Technique

- Finding redundant constraints applied to graph problem

Conclusion

q-Coloring has

- a kernel with bitsize $O(k^{q-1} \log k)$
- no kernel with bitsize $O(k^{q-1-\varepsilon})$ unless NP $\subseteq$ coNP/poly

Technique

- Finding redundant constraints applied to graph problem

Conclusion

q-Coloring has

- a kernel with bitsize $O(k^{q-1} \log k)$
- no kernel with bitsize $O(k^{q-1-\varepsilon})$ unless NP $\subseteq$ coNP/poly

Technique

- Finding redundant constraints applied to graph problem

Thank you!

Conclusion

q-Coloring has

- a kernel with bitsize $O(k^{q-1} \log k)$
- no kernel with bitsize $O(k^{q-1-\varepsilon})$ unless NP $\subseteq$ coNP/poly

Technique

- Finding redundant constraints applied to graph problem

Thank you!

Conclusion

q-Coloring has

- a kernel with bitsize $O(k^{q-1} \log k)$
- no kernel with bitsize $O(k^{q-1-\varepsilon})$ unless NP $\subseteq$ coNP/poly

Technique

- Finding redundant constraints applied to graph problem

Thank you!

Polynomial for q-Coloring

For 3-Coloring:

$$a \wedge d + a \wedge e + d \wedge e +$$
$$d \wedge a + e \wedge a + e \wedge d \equiv_2 0$$

Formula:

$$y_{1,1} \cdot y_{2,2} + y_{1,1} \cdot y_{3,2} + y_{2,1} \cdot y_{1,2} + y_{2,1} \cdot y_{3,2} + y_{3,1} \cdot y_{1,2} + y_{3,1} \cdot y_{2,2} \equiv_2 0$$

In general:

$$\sum_{\substack{i_1,\ldots,i_{q-1} \in [q] \\ \text{distinct}}} \prod_{k=1}^{q-1} y_{i_k,k} \equiv_2 0$$