Dual parameterization of WEIGHTED COLORING

Júlio Araújo[1] Victor A. Campos[2] Carlos Vinícius G. C. Lima[3]
Vinícius F. Dos Santos[3] Ignasi Sau[4] Ana Silva[1]

IPEC 2018, Helsinki, Finland
August 21-24, 2018

[1] Departamento de Matemática, Universidade Federal do Ceará, **Fortaleza**, **Brazil**.

[2] Departamento de Computação, Universidade Federal do Ceará, **Fortaleza**, **Brazil**.

[3] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, **Belo Horizonte**, **Brazil**.

[4] CNRS, LIRMM, Université de Montpellier, **Montpellier**, **France**.

Outline of the talk

1. Introduction
2. Our results
3. Sketches of some proofs
4. Conclusions

Next section is...

1 Introduction

2 Our results

3 Sketches of some proofs

4 Conclusions

Weighted Coloring

We are given a graph G together with a weight function $w : V(G) \to \mathbb{R}^+$.

A (proper) k-coloring of G is a partition $c = (S_i)_{i \in [1,k]}$ of $V(G)$ into k stable sets $S_1, \ldots, S_k$.

Weighted Coloring

We are given a graph G together with a weight function $w : V(G) \to \mathbb{R}^+$.

A (proper) k-coloring of G is a partition $c = (S_i)_{i \in [1,k]}$ of $V(G)$ into k stable sets $S_1, \dots, S_k$.

The weight of a color S_i is $w(i) = \max_{v \in S_i} w(v)$.

Weighted Coloring

We are given a graph G together with a weight function $w : V(G) \to \mathbb{R}^+$.

A (proper) k-coloring of G is a partition $c = (S_i)_{i \in [1,k]}$ of $V(G)$ into k stable sets $S_1, \ldots, S_k$.

The weight of a color S_i is $w(i) = \max_{v \in S_i} w(v)$.

The weight of a coloring c is $w(c) = \sum_{i=1}^{k} w(i)$.

Weighted Coloring

We are given a graph G together with a weight function $w : V(G) \to \mathbb{R}^+$.

A (proper) k-coloring of G is a partition $c = (S_i)_{i \in [1,k]}$ of $V(G)$ into k stable sets $S_1, \ldots, S_k$.

The weight of a color S_i is $w(i) = \max_{v \in S_i} w(v)$.

The weight of a coloring c is $w(c) = \sum_{i=1}^{k} w(i)$.

The weighted chromatic number of a pair (G, w) is

$$\sigma(G, w) \; = \; \min\{w(c) \mid c \text{ is a proper coloring of } G\}.$$

Weighted Coloring

We are given a graph G together with a weight function $w : V(G) \to \mathbb{R}^+$.

A (proper) k-coloring of G is a partition $c = (S_i)_{i \in [1,k]}$ of $V(G)$ into k stable sets $S_1, \ldots, S_k$.

The weight of a color S_i is $w(i) = \max_{v \in S_i} w(v)$.

The weight of a coloring c is $w(c) = \sum_{i=1}^{k} w(i)$.

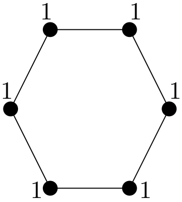
The weighted chromatic number of a pair (G, w) is

$$\sigma(G, w) \;=\; \min\{w(c) \mid c \text{ is a proper coloring of } G\}.$$

If $w(v) = 1$ for every $v \in V(G)$, then clearly $\sigma(G, w) = \chi(G)$.

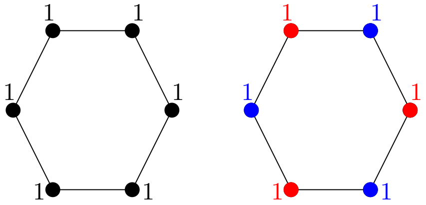
Example

(G, w)



Example

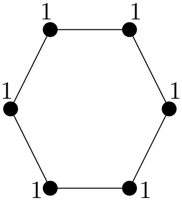
(G, w)

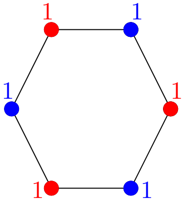


$\sigma(G, w) = 2$

Example

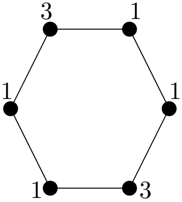
(G, w)





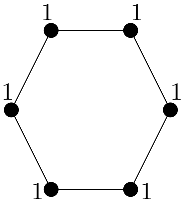
$\sigma(G, w) = 2$

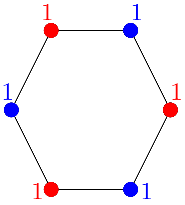
(G, w')



Example

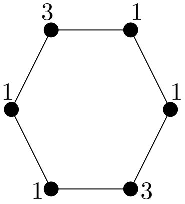
(G, w)

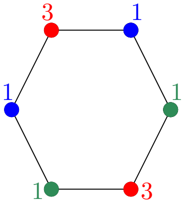




$\sigma(G, w) = 2$

(G, w')





$\sigma(G, w') = 5$

What is known about Weighted Coloring

The Weighted Coloring problem was introduced by [Guan, Zhu. 1997] to study practical applications related to resource allocation.

If all the vertex weights are equal to one, then $\sigma(G, w) = \chi(G)$.
Thus, determining $\sigma(G, w)$ in an NP-hard problem.

What is known about WEIGHTED COLORING

The WEIGHTED COLORING problem was introduced by [Guan, Zhu. 1997] to study practical applications related to resource allocation.

If all the vertex weights are equal to one, then $\sigma(G, w) = \chi(G)$.
Thus, determining $\sigma(G, w)$ in an NP-hard problem.

The problem is NP-hard even on:

- split graphs, interval graphs, bipartite graphs, and triangle-free planar graphs with bounded degree.

On the other hand, it is polynomial on

- cographs and some subclasses of bipartite graphs.

[de Werra, Demange, Monnot, Paschos. 2002]
[Escoffier, Monnot, Paschos. 2006]
[de Werra, Demange, Escoffier, Monnot, Paschos. 2009]

Complexity of WEIGHTED COLORING on trees (forests)

WEIGHTED COLORING can be solved on forests in time $n^{\mathcal{O}(\log n)} = 2^{\mathcal{O}(\log^2 n)}$ (quasi-polynomial). [Guan, Zhu. 1997] [Linhares, Reed. 2006]

Complexity of Weighted Coloring on trees (forests)

Weighted Coloring can be solved on forests in time

$n^{\mathcal{O}(\log n)} = 2^{\mathcal{O}(\log^2 n)}$ (quasi-polynomial). [Guan, Zhu. 1997]

[Linhares, Reed. 2006]

Is Weighted Coloring **polynomial** on trees/forests?

More generally, on graphs of bounded treewidth?

Complexity of WEIGHTED COLORING on trees (forests)

WEIGHTED COLORING can be solved on forests in time $n^{\mathcal{O}(\log n)} = 2^{\mathcal{O}(\log^2 n)}$ (quasi-polynomial). [Guan, Zhu. 1997]
[Linhares, Reed. 2006]

Is WEIGHTED COLORING **polynomial** on trees/forests?
More generally, on graphs of bounded treewidth?

- PTAS on bounded treewidth graphs. [Escoffier, Monnot, Paschos. 2006]
- Polynomial on the class of trees where vertices with degree at least three induce a stable set. [Kavitha, Mestre. 2012]

Complexity of WEIGHTED COLORING on trees (forests)

WEIGHTED COLORING can be solved on forests in time $n^{\mathcal{O}(\log n)} = 2^{\mathcal{O}(\log^2 n)}$ (quasi-polynomial). [Guan, Zhu. 1997] [Linhares, Reed. 2006]

Is WEIGHTED COLORING **polynomial** on trees/forests?
More generally, on graphs of bounded treewidth?

- PTAS on bounded treewidth graphs. [Escoffier, Monnot, Paschos. 2006]
- Polynomial on the class of trees where vertices with degree at least three induce a stable set. [Kavitha, Mestre. 2012]

★ There is no algorithm solving WEIGHTED COLORING on n-vertex trees in time $n^{o(\log n)}$, under the ETH. [Araújo, Nisse, Pérennes. 2014]

Complexity of Weighted Coloring on trees (forests)

Weighted Coloring can be solved on forests in time
$n^{\mathcal{O}(\log n)} = 2^{\mathcal{O}(\log^2 n)}$ (quasi-polynomial). [Guan, Zhu. 1997]
[Linhares, Reed. 2006]

Is Weighted Coloring **polynomial** on trees/forests?
More generally, on graphs of bounded treewidth?

- PTAS on bounded treewidth graphs. [Escoffier, Monnot, Paschos. 2006]
- Polynomial on the class of trees where vertices with degree at least three induce a stable set. [Kavitha, Mestre. 2012]

★ There is no algorithm solving Weighted Coloring on n-vertex trees in time $n^{o(\log n)}$, under the ETH. [Araújo, Nisse, Pérennes. 2014]

Weighted Coloring on forests is W[1]-hard parameterized by the size of a largest connected component. [Araújo, Baste, S. 2017]

Dual parameterization: saving colors

$\boxed{\text{Conclusion}}$ WEIGHTED COLORING is a very hard problem!

Dual parameterization: saving colors

Conclusion WEIGHTED COLORING is a very hard problem!

Let's try to give some good news:

DUAL WEIGHTED COLORING
Input: A vertex-weighted graph (G, w) and a positive integer k.
Parameter: k.
Question: Is $\sigma(G, w) \leq \sum_{v \in V(G)} w(v) - k$?

Dual parameterization: saving colors

$\boxed{\text{Conclusion}}$ WEIGHTED COLORING is a very hard problem!

Let's try to give some good news:

DUAL WEIGHTED COLORING **Input:** A vertex-weighted graph (G, w) and a positive integer k. **Parameter:** k. **Question:** Is $\sigma(G, w) \leq \sum_{v \in V(G)} w(v) - k$?

(We assume that all vertex-weights are positive integers.)

Dual parameterization: saving colors

Conclusion WEIGHTED COLORING is a very hard problem!

Let's try to give some good news:

DUAL WEIGHTED COLORING **Input:** A vertex-weighted graph (G, w) and a positive integer k. **Parameter:** k. **Question:** Is $\sigma(G, w) \leq \sum_{v \in V(G)} w(v) - k$?

(We assume that all vertex-weights are positive integers.)

The dual parameterization has proved to be useful for VERTEX COLORING, GRUNDY COLORING, and b-COLORING.

[Chor, Fellows, Juedes. 2004]

[Havet, Sampaio. 2013]

Known results

Dual Weighted Coloring has been recently studied: [Escoffier. 2016]

Known results

DUAL WEIGHTED COLORING has been recently studied: [Escoffier. 2016]

WARNING!! WEIGHTED COLORING is also called MAX COLORING in the literature.

Known results

DUAL WEIGHTED COLORING has been recently studied: [Escoffier. 2016]

WARNING!! WEIGHTED COLORING is also called MAX COLORING in the literature.

Escoffier showed that the problem is FPT, namely:

- Algorithm running in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$.

Known results

DUAL WEIGHTED COLORING has been recently studied: [Escoffier. 2016]

WARNING!! WEIGHTED COLORING is also called MAX COLORING in the literature.

Escoffier showed that the problem is FPT, namely:

- Algorithm running in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$.
- Asked "what size of kernel can be achieved for the problem?"

Next section is...

1 Introduction

2 Our results

3 Sketches of some proofs

4 Conclusions

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]
   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]
   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1}+1)(k-1)$ vertices.

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]
   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1}+1)(k-1)$ vertices.
   No polynomial kernels unless NP $\subseteq$ coNP/poly,
   even on split graphs with only two different weights.

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.

   Inspired from an algorithm for the chromatic number. [Lawler. 1976]

   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1} + 1)(k - 1)$ vertices.

   No polynomial kernels unless NP $\subseteq$ coNP/poly,
   even on split graphs with only two different weights.

   (Contrast with Dual Vertex Coloring, which admits a linear kernel.)

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.

   Inspired from an algorithm for the chromatic number. [Lawler. 1976]

   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1} + 1)(k-1)$ vertices.

   No polynomial kernels unless NP $\subseteq$ coNP/poly,
   even on split graphs with only two different weights.

   (Contrast with Dual Vertex Coloring, which admits a linear kernel.)

3. Polynomial kernels on particular graph classes:

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]
   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1}+1)(k-1)$ vertices.
   No polynomial kernels unless NP $\subseteq$ coNP/poly,
   even on split graphs with only two different weights.
   (Contrast with Dual Vertex Coloring, which admits a linear kernel.)

3. Polynomial kernels on particular graph classes:
   - Linear kernel on graph classes with bounded clique number.

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]
   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1}+1)(k-1)$ vertices.
   No polynomial kernels unless $\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{poly}$,
   even on split graphs with only two different weights.
   (Contrast with Dual Vertex Coloring, which admits a linear kernel.)

3. Polynomial kernels on particular graph classes:
   - Linear kernel on graph classes with bounded clique number.
   - Cubic kernel on interval graphs.
     Based on the "consecutive ones property" of interval graphs.

Our results for Dual Weighted Coloring

1. FPT algorithm running in time $9^k \cdot n^{\mathcal{O}(1)}$.
   Inspired from an algorithm for the chromatic number. [Lawler. 1976]
   No algorithm in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ exists under the ETH.

2. Kernel with at most $(2^{k-1}+1)(k-1)$ vertices.
   No polynomial kernels unless NP $\subseteq$ coNP/poly,
   even on split graphs with only two different weights.
   (Contrast with Dual Vertex Coloring, which admits a linear kernel.)

3. Polynomial kernels on particular graph classes:
   - Linear kernel on graph classes with bounded clique number.
   - Cubic kernel on interval graphs.
     Based on the "consecutive ones property" of interval graphs.
   - Subclasses of split graphs, with lower bounds on the degrees.

Next section is...

1 Introduction

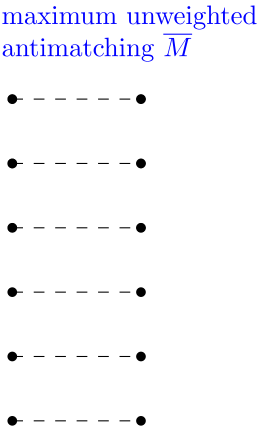
2 Our results

3 Sketches of some proofs

4 Conclusions

We start with an idea already used, in particular, by [Escoffier. 2016]

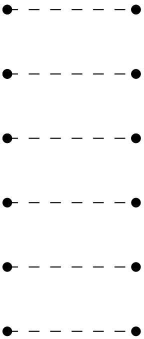
An exponential kernel for Dual Weighted Coloring



We compute (in poly time) a maximum unweighted antimatching $\overline{M}$.

An exponential kernel for DUAL WEIGHTED COLORING

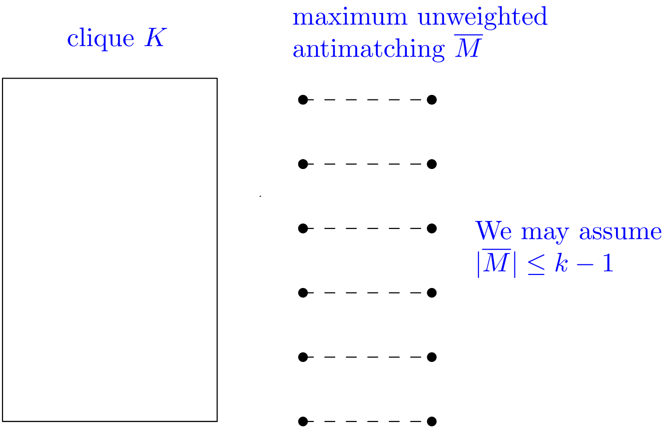
maximum unweighted antimatching $\overline{M}$



We may assume $|\overline{M}| \leq k - 1$

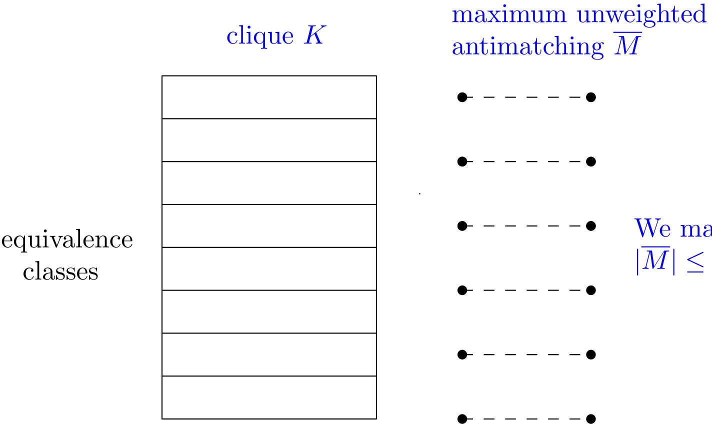
If $|\overline{M}| \geq k$, we are dealing with a YES-instance.

An exponential kernel for DUAL WEIGHTED COLORING



The complement of the antimatching $\overline{M}$ induces a clique K.

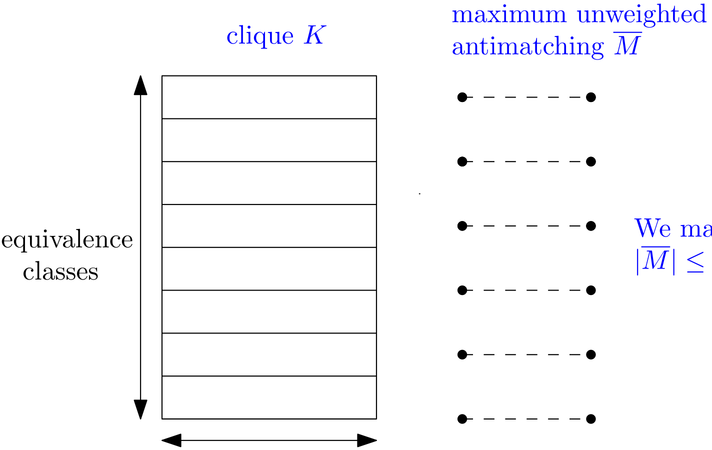
An exponential kernel for DUAL WEIGHTED COLORING



We may assume $|\overline{M}| \leq k - 1$

We partition K into equivalence classes w.r.t. the neighborhood in $\overline{M}$.

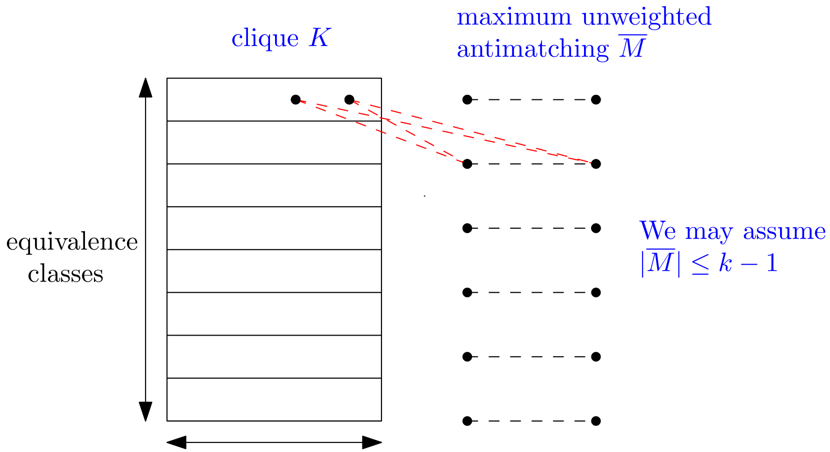
An exponential kernel for Dual Weighted Coloring



We may assume $|\overline{M}| \leq k - 1$

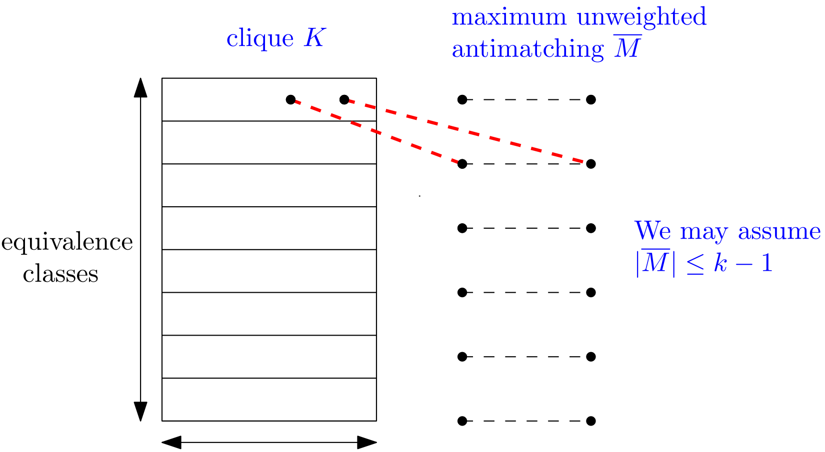
We have to bound the number and the size of the equivalence classes.

An exponential kernel for DUAL WEIGHTED COLORING



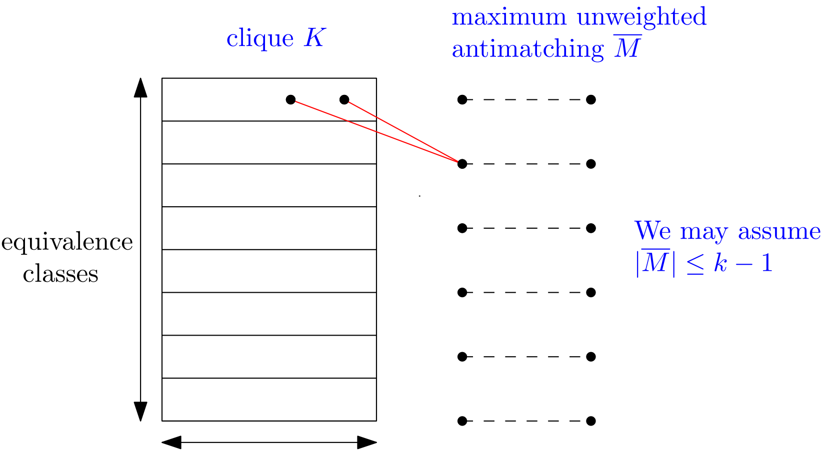
An equivalence class cannot miss both endvertices of a non-edge.

An exponential kernel for Dual Weighted Coloring



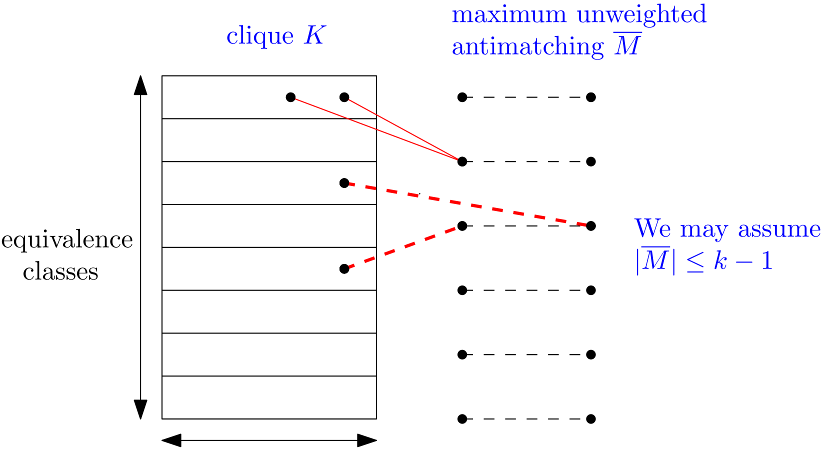
An equivalence class cannot miss both endvertices of a non-edge.

An exponential kernel for Dual Weighted Coloring



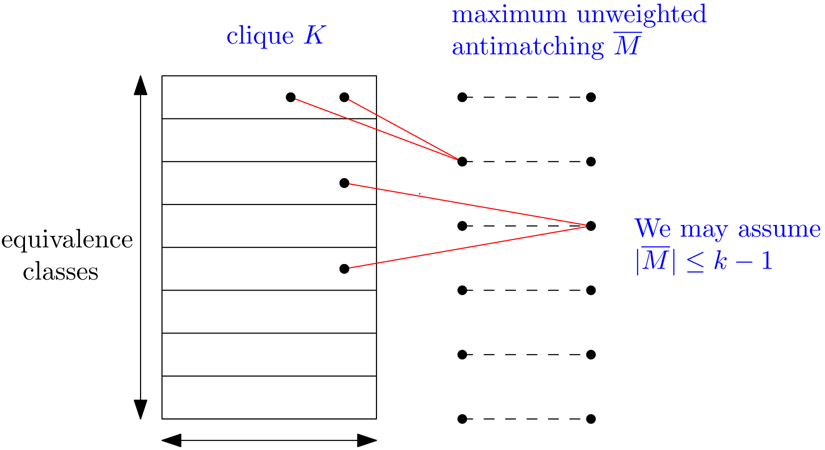
An equivalence class cannot miss both endvertices of a non-edge.

An exponential kernel for Dual Weighted Coloring



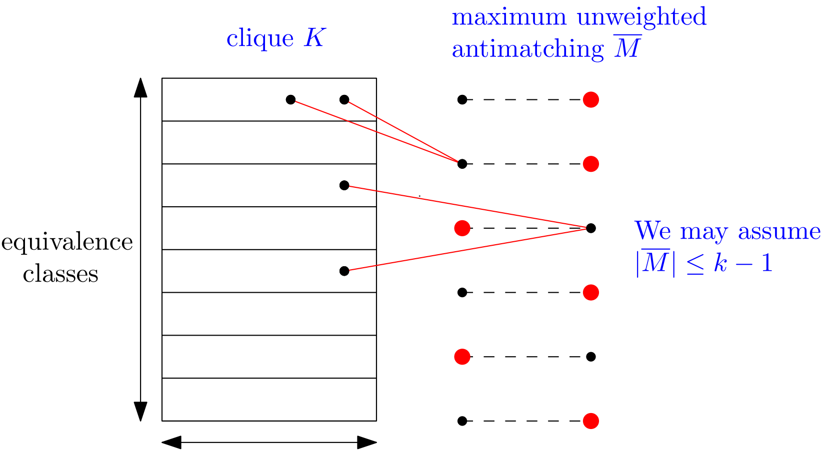
Two equivalence classes cannot miss different endvertices of a non-edge.

An exponential kernel for Dual Weighted Coloring



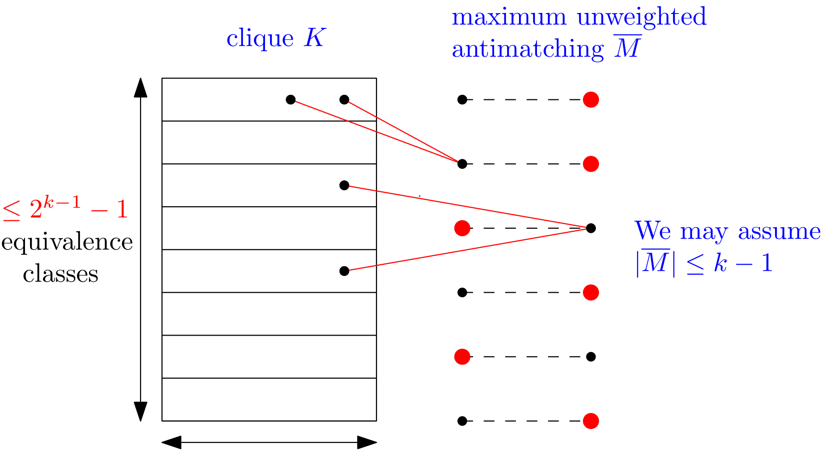
Two equivalence classes cannot miss different endvertices of a non-edge.

An exponential kernel for Dual Weighted Coloring



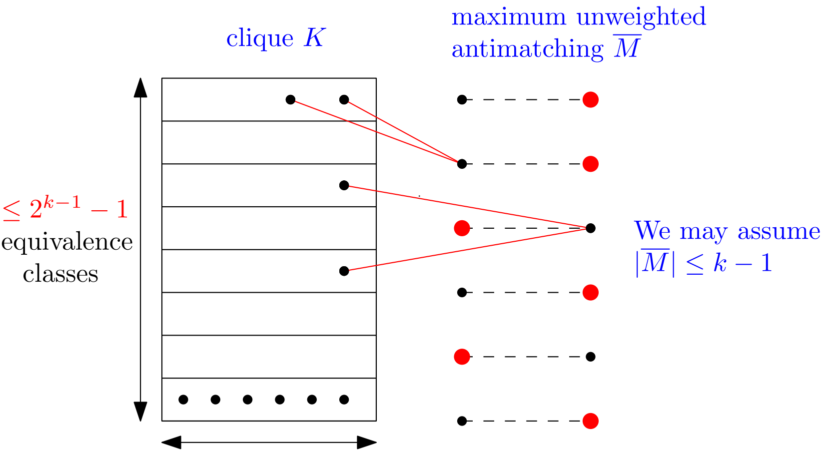
Every non-edge has a unique "potential non-neighbor" for the classes.

An exponential kernel for Dual Weighted Coloring



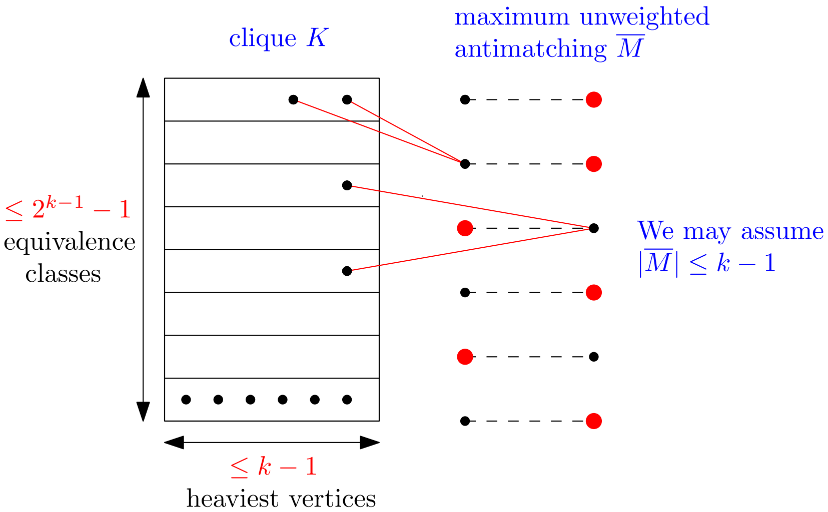
Hence, the number of equivalence classes is at most $2^{k-1} - 1$.

An exponential kernel for Dual Weighted Coloring



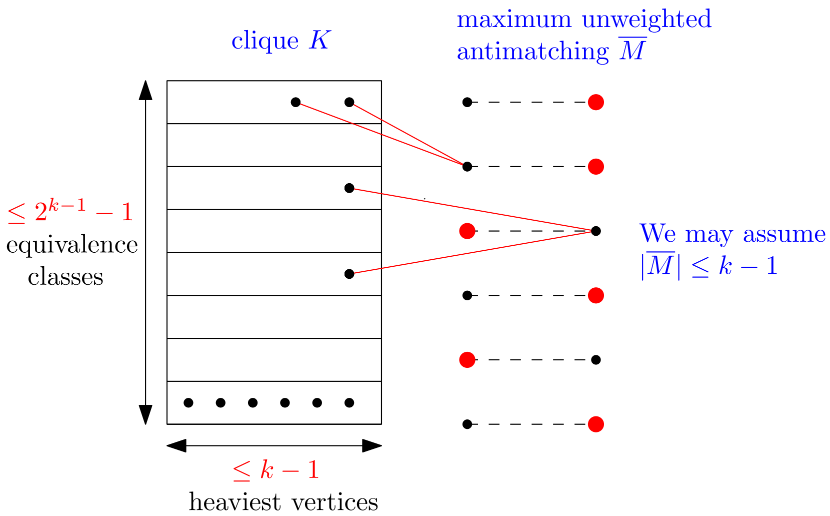
It remains to bound the size of each equivalence class.

An exponential kernel for Dual Weighted Coloring



It is safe to keep only the $k-1$ heaviest vertices of each equivalence class.

An exponential kernel for Dual Weighted Coloring



Thus, $|V(G)| \leq |V(\overline{M})| + |K| \ \leq \ 2(k-1) + (2^{k-1}-1)(k-1)$.

Ruling out polynomial kernels

Theorem

DUAL WEIGHTED COLORING *does not admit a polynomial kernel unless* NP $\subseteq$ coNP/poly*, even on split graphs with only two different weights.*

Ruling out polynomial kernels

Theorem

Dual Weighted Coloring *does not admit a polynomial kernel unless* NP $\subseteq$ coNP/poly, *even on split graphs with only two different weights.*

We present a polynomial parameter transformation from Set Cover parameterized by the size of the universe, known not to admit polynomial kernels unless NP $\subseteq$ coNP/poly. [Dom, Lokshtanov, Saurabh. 2014]

Ruling out polynomial kernels

Theorem

Dual Weighted Coloring *does not admit a polynomial kernel unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, *even on split graphs with only two different weights.*

We present a polynomial parameter transformation from Set Cover parameterized by the size of the universe, known not to admit polynomial kernels unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. [Dom, Lokshtanov, Saurabh. 2014]

Let $A, B \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems.

A polynomial parameter transformation from A to B is an algorithm s.t.:

Instance (x, k) of A $\xrightarrow{\text{polynomial time}}$ Instance (x', k') of B

1. (x, k) is a Yes-instance of $A \Leftrightarrow (x', k')$ is a Yes-instance of B.
2. $k' \leq \text{poly}(k)$.

[Bodlaender, Thomassé, Yeo. 2011]

Ruling out polynomial kernels

Theorem

Dual Weighted Coloring *does not admit a polynomial kernel unless* $NP \subseteq coNP/poly$*, even on split graphs with only two different weights.*

We present a polynomial parameter transformation from Set Cover parameterized by the size of the universe, known not to admit polynomial kernels unless $NP \subseteq coNP/poly$. [Dom, Lokshtanov, Saurabh. 2014]

Our reduction is an appropriate modification of a reduction to prove the NP-hardness of Weighted Coloring on split graphs:

only the weights change.

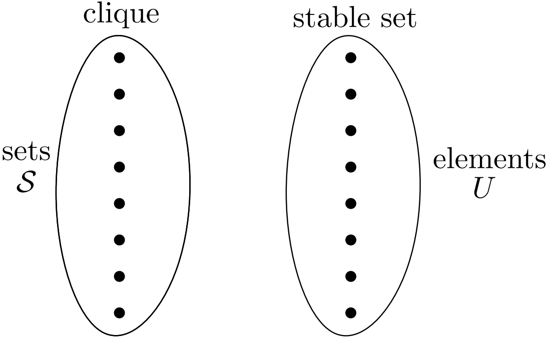
[Demange, de Werra, Monnot, Paschos. 2002]

Polynomial parameter transformation from Set Cover

We are given an instance $(U, \mathcal{S}, k, \ell)$ of Set Cover,
where $\mathcal{S}$ is a family of sets over a universe U of size k (the parameter).
Question: $\exists$? $\mathcal{S}' \subseteq \mathcal{S}$ of at most ℓ sets covering all the elements of U.

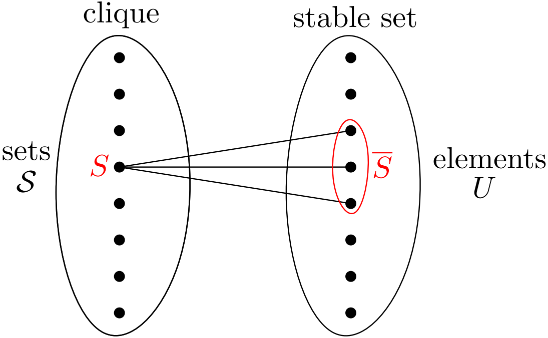
Polynomial parameter transformation from SET COVER

We are given an instance $(U, \mathcal{S}, k, \ell)$ of SET COVER,
where $\mathcal{S}$ is a family of sets over a universe U of size k (the parameter).
Question: $\exists ?$ $\mathcal{S}' \subseteq \mathcal{S}$ of at most ℓ sets covering all the elements of U.



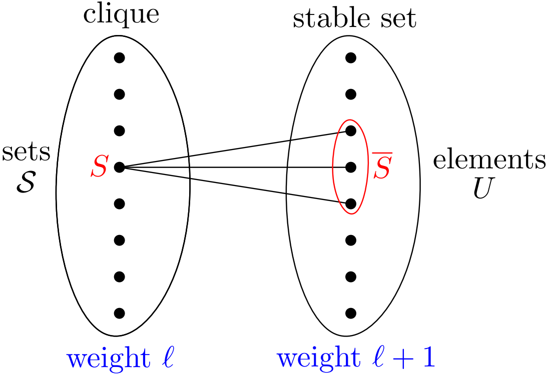
Polynomial parameter transformation from Set Cover

We are given an instance $(U, \mathcal{S}, k, \ell)$ of Set Cover,
where $\mathcal{S}$ is a family of sets over a universe U of size k (the parameter).
Question: $\exists ?\ \mathcal{S}' \subseteq \mathcal{S}$ of at most ℓ sets covering all the elements of U.



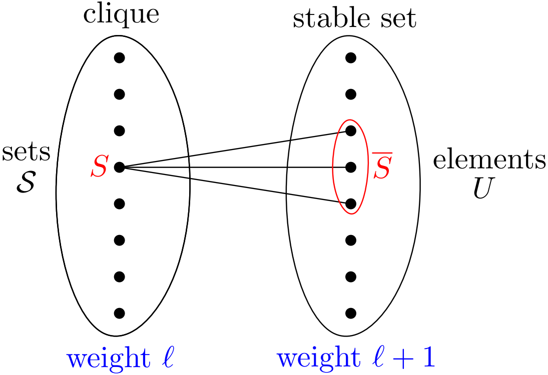
Polynomial parameter transformation from Set Cover

We are given an instance $(U, \mathcal{S}, k, \ell)$ of Set Cover,
where $\mathcal{S}$ is a family of sets over a universe U of size k (the parameter).
Question: $\exists?$ $\mathcal{S}' \subseteq \mathcal{S}$ of at most ℓ sets covering all the elements of U.



Polynomial parameter transformation from SET COVER

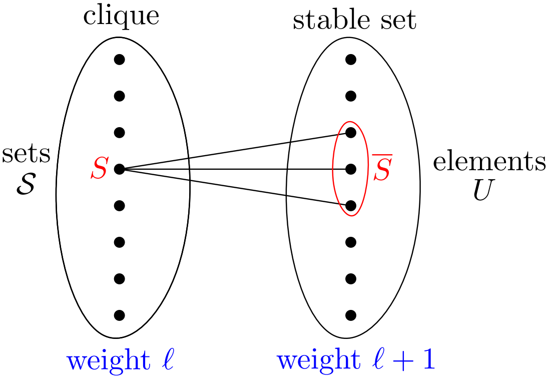
We are given an instance $(U, \mathcal{S}, k, \ell)$ of SET COVER,
where $\mathcal{S}$ is a family of sets over a universe U of size k (the parameter).
Question: $\exists? \ \mathcal{S}' \subseteq \mathcal{S}$ of at most ℓ sets covering all the elements of U.



We build and instance (G, w, k'), with $k' = k(\ell + 1) - \ell = \mathcal{O}(k^2)$.

Polynomial parameter transformation from Set Cover

We are given an instance $(U, \mathcal{S}, k, \ell)$ of Set Cover,
where $\mathcal{S}$ is a family of sets over a universe U of size k (the parameter).
Question: $\exists$? $\mathcal{S}' \subseteq \mathcal{S}$ of at most ℓ sets covering all the elements of U.



$(U, \mathcal{S}, k, \ell)$ satisfiable $\Leftrightarrow$ $\sigma(G, w) \leq \sum_{v \in V(G)} w(v) - k' = |\mathcal{S}| \cdot \ell + \ell$.

Dual Weighted Coloring seems to be strongly related to Set Cover...

Exploiting the relation with Set Cover

Dual Weighted Coloring seems to be strongly related to Set Cover... and also to d-Set Cover:

Proposition

Dual Weighted Coloring restricted to split graphs such that each vertex in the clique has at most d non-neighbors in the stable set, for some constant $d \geq 2$, admits a kernel with $\mathcal{O}(k^d)$ vertices.

Exploiting the relation with SET COVER

DUAL WEIGHTED COLORING seems to be strongly related to SET COVER... and also to d-SET COVER:

Proposition

DUAL WEIGHTED COLORING *restricted to split graphs such that each vertex in the clique has at most d non-neighbors in the stable set, for some constant $d \geq 2$, admits a kernel with $\mathcal{O}(k^d)$ vertices.*

Furthermore, for any $\varepsilon > 0$, a kernel with $\mathcal{O}(k^{\frac{d-3}{2}-\varepsilon})$ vertices does not exist unless $\mathsf{NP} \subseteq \mathsf{coNP}/\mathsf{poly}$.

Using the kernel lower bound for d-SET COVER. [Hermelin, Wu. 2012]

Next section is...

1 Introduction

2 Our results

3 Sketches of some proofs

4 Conclusions

Some questions for further research

- We presented an FPT algorithm for Dual Weighted Coloring running in time $9^k \cdot n^{\mathcal{O}(1)}$.

Some questions for further research

- We presented an FPT algorithm for Dual Weighted Coloring running in time $9^k \cdot n^{\mathcal{O}(1)}$.

  Improve the running time and/or prove lower bounds under the SETH.

Some questions for further research

- We presented an FPT algorithm for DUAL WEIGHTED COLORING running in time $9^k \cdot n^{\mathcal{O}(1)}$.

  Improve the running time and/or prove lower bounds under the SETH.

- Can the cubic kernel on interval graphs be improved?

Some questions for further research

- We presented an FPT algorithm for Dual Weighted Coloring running in time $9^k \cdot n^{\mathcal{O}(1)}$.

  Improve the running time and/or prove lower bounds under the SETH.

- Can the cubic kernel on interval graphs be improved?

- Identify other classes of (dense) graphs allowing for polynomial kernels.

Gràcies!

LLIBERTAT
PRESOS POLÍTICS

FREEDOM FOR ALL CATALAN POLITICAL PRISONERS IN SPAIN