Ruling out FPT algorithms for WEIGHTED COLORING on forests

Júlio Araújo[1] Julien Baste[2] Ignasi Sau[1,2]

LAGOS, CIRM, Marseille, France
September 14, 2017

Full version available at [arXiv:1703.09726]

[1] Departamento de Matemática, UFC, Fortaleza, Brazil.

[2] CNRS, LIRMM, Université de Montpellier, Montpellier, France.

Outline of the talk

1. Introduction
2. Our results
3. Some ideas of the proofs

Next section is...

1. Introduction
2. Our results
3. Some ideas of the proofs

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 $S_1, \ldots, S_k$.

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 $S_1, \dots, S_k$.

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

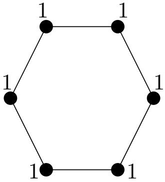
$$\sigma(G, w) \ = \ \min\{w(c) \mid c \text{ is a proper coloring of } G\}.$$

For a positive integer r, we define

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

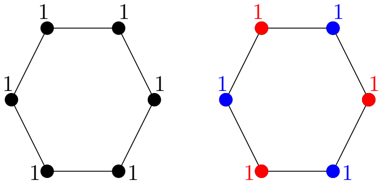
Example

(G, w)



Example

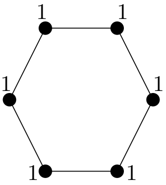
(G, w)

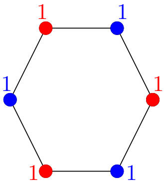


$\sigma(G, w) = 2$

Example

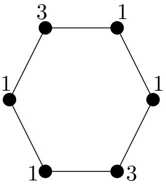
(G, w)



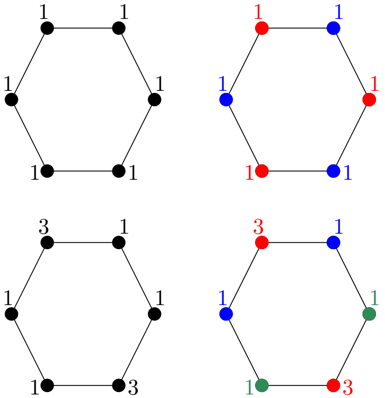


$\sigma(G, w) = 2$

(G, w')



Example



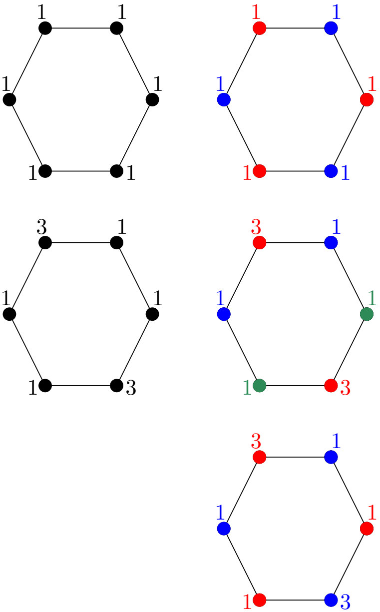
(G, w) $\sigma(G, w) = 2$

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

Example

(G, w)

(G, w')



$\sigma(G, w) = 2$

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

$\sigma(G, w'; 2) = 6$

What is known about WEIGHTED COLORING

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

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 $\sigma(G, w)$ and $\sigma(G, w; r)$ are NP-hard problems.

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 $\sigma(G, w)$ and $\sigma(G, w; r)$ are NP-hard problems.

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 (or forests)

Complexity of weighted coloring on trees (or forests)

On an n-vertex graph of treewidth t, $\sigma(G, w; r)$ can be computed in time

$$n^{O(r)} \cdot r^{O(t)}.$$

[Guan, Zhu. 1997]

Complexity of weighted coloring on trees (or forests)

On an n-vertex graph of treewidth t, $\sigma(G, w; r)$ can be computed in time

$$n^{O(r)} \cdot r^{O(t)}.$$

[Guan, Zhu. 1997]

They showed that we may assume $r \leq \chi_{\mathsf{FF}}(G)$ (first-fit chromatic number).

Complexity of weighted coloring on trees (or forests)

On an n-vertex graph of treewidth t, $\sigma(G, w; r)$ can be computed in time

$$n^{O(r)} \cdot r^{O(t)}.$$

[Guan, Zhu. 1997]

They showed that we may assume $r \leq \chi_{\mathsf{FF}}(G)$ (first-fit chromatic number).

For any graph G, it holds that $\chi_{\mathsf{FF}}(G) = O(t \log n)$. [Linhares, Reed. 2006]

Complexity of weighted coloring on trees (or forests)

On an n-vertex graph of treewidth t, $\sigma(G, w; r)$ can be computed in time

$$n^{O(r)} \cdot r^{O(t)}.$$

[Guan, Zhu. 1997]

They showed that we may assume $r \leq \chi_{\mathsf{FF}}(G)$ (first-fit chromatic number).

For any graph G, it holds that $\chi_{\mathsf{FF}}(G) = O(t \log n)$. [Linhares, Reed. 2006]

$\Longrightarrow$ Weighted Coloring can be solved on forests in time $n^{O(\log n)} = 2^{O(\log^2 n)}$ (quasi-polynomial).

Complexity of weighted coloring on trees (or forests)

On an n-vertex graph of treewidth t, $\sigma(G, w; r)$ can be computed in time

$$n^{O(r)} \cdot r^{O(t)}.$$

[Guan, Zhu. 1997]

They showed that we may assume $r \leq \chi_{\mathsf{FF}}(G)$ (first-fit chromatic number).

For any graph G, it holds that $\chi_{\mathsf{FF}}(G) = O(t \log n)$. [Linhares, Reed. 2006]

$\Longrightarrow$ Weighted Coloring can be solved on forests in time
$n^{O(\log n)} = 2^{O(\log^2 n)}$ (quasi-polynomial).

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

Complexity of weighted coloring on trees (or forests)

On an n-vertex graph of treewidth t, $\sigma(G, w; r)$ can be computed in time

$$n^{O(r)} \cdot r^{O(t)}.$$

[Guan, Zhu. 1997]

They showed that we may assume $r \leq \chi_{FF}(G)$ (first-fit chromatic number).

For any graph G, it holds that $\chi_{FF}(G) = O(t \log n)$. [Linhares, Reed. 2006]

$\Longrightarrow$ WEIGHTED COLORING can be solved on forests in time $n^{O(\log n)} = 2^{O(\log^2 n)}$ (quasi-polynomial).

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

Some partial results:

- 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]

The problem has been recently solved!

The question of [Guan, Zhu. 1997] has been answered only recently:

Theorem (Araújo, Nisse, Pérennes. 2014)

Unless the ETH fails, there is no algorithm computing the weighted chromatic number of n-vertex trees in time $n^{o(\log n)}$.

The problem has been recently solved!

The question of [Guan, Zhu. 1997] has been answered only recently:

Theorem (Araújo, Nisse, Pérennes. 2014)

Unless the ETH fails, there is no algorithm computing the weighted chromatic number of n-vertex trees in time $n^{o(\log n)}$.

Exponential Time Hypothesis (ETH): the 3-SAT problem on formulas with n variables cannot be solved in subexponential time, that is, $2^{o(n)}$.

[Impagliazzo, Paturi, Zane. 2001]

The problem has been recently solved!

The question of [Guan, Zhu. 1997] has been answered only recently:

Theorem (Araújo, Nisse, Pérennes. 2014)

Unless the ETH fails, there is no algorithm computing the weighted chromatic number of n-vertex trees in time $n^{o(\log n)}$.

Exponential Time Hypothesis (ETH): the 3-SAT problem on formulas with n variables cannot be solved in subexponential time, that is, $2^{o(n)}$.

[Impagliazzo, Paturi, Zane. 2001]

That is, the running time $n^{O(\log n)}$ is tight under the ETH.

The problem has been recently solved!

The question of [Guan, Zhu. 1997] has been answered only recently:

Theorem (Araújo, Nisse, Pérennes. 2014)

Unless the ETH fails, there is no algorithm computing the weighted chromatic number of n-vertex trees in time $n^{o(\log n)}$.

Exponential Time Hypothesis (ETH): the 3-SAT problem on formulas with n variables cannot be solved in subexponential time, that is, $2^{o(n)}$.

[Impagliazzo, Paturi, Zane. 2001]

That is, the running time $n^{O(\log n)}$ is tight under the ETH.

- Weighted Coloring on forests is unlikely to be in P, as this would contradict the ETH.

The problem has been recently solved!

The question of [Guan, Zhu. 1997] has been answered only recently:

Theorem (Araújo, Nisse, Pérennes. 2014)

Unless the ETH fails, there is no algorithm computing the weighted chromatic number of n-vertex trees in time $n^{o(\log n)}$.

Exponential Time Hypothesis (ETH): the 3-SAT problem on formulas with n variables cannot be solved in subexponential time, that is, $2^{o(n)}$.

[Impagliazzo, Paturi, Zane. 2001]

That is, the running time $n^{O(\log n)}$ is tight under the ETH.

- Weighted Coloring on forests is unlikely to be in P, as this would contradict the ETH.
- Also unlikely to be NP-hard, as all problems in NP could be solved in subexponential time, contradicting again the ETH.

Can we relax the complexity hypothesis?

Can we relax the complexity hypothesis?

Objective of this article

Providing hardness results for computing $\sigma(G, w)$ and $\sigma(G, w; r)$ when G is a forest, relying on complexity assumptions weaker than the ETH.

Can we relax the complexity hypothesis?

Objective of this article

Providing hardness results for computing $\sigma(G, w)$ and $\sigma(G, w; r)$ when G is a forest, relying on complexity assumptions weaker than the ETH.

We study the problem from the viewpoint of parameterized complexity, and we assume the weaker hypothesis $\mathrm{FPT} \neq \mathrm{W}[1]$.

Can we relax the complexity hypothesis?

Objective of this article

Providing hardness results for computing $\sigma(G, w)$ and $\sigma(G, w; r)$ when G is a forest, relying on complexity assumptions weaker than the ETH.

We study the problem from the viewpoint of parameterized complexity, and we assume the weaker hypothesis $\mathrm{FPT} \neq \mathrm{W}[1]$.

Indeed, it is well-known that

$$\boxed{\mathrm{ETH}} \implies \boxed{\mathrm{FPT} \neq \mathrm{W}[1]} \implies \boxed{\mathrm{P} \neq \mathrm{NP}}$$

Can we relax the complexity hypothesis?

Objective of this article

Providing hardness results for computing $\sigma(G, w)$ and $\sigma(G, w; r)$ when G is a forest, relying on complexity assumptions weaker than the ETH.

We study the problem from the viewpoint of parameterized complexity, and we assume the weaker hypothesis FPT $\neq$ W[1].

Indeed, it is well-known that

$$\boxed{\text{ETH}} \implies \boxed{\text{FPT} \neq \text{W[1]}} \implies \boxed{\text{P} \neq \text{NP}}$$

A few words on parameterized complexity

Instances of a parameterized problem: come with an integer parameter k.

A few words on parameterized complexity

Instances of a parameterized problem: come with an integer parameter k.

A parameterized problem is fixed-parameter tractable (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ solves the problem in time bounded by $f(k) \cdot |I|^c$.

A few words on parameterized complexity

Instances of a parameterized problem: come with an integer parameter k.

A parameterized problem is fixed-parameter tractable (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ solves the problem in time bounded by $f(k) \cdot |I|^c$.

Parameterized reduction: given an input $I = (x, k)$ of the source problem, computes in time $f(k) \cdot |I|^c$, an equivalent instance $I' = (x', k')$ of the target problem, such that $k' \leq g(k)$ for some function g.

A few words on parameterized complexity

Instances of a parameterized problem: come with an integer parameter k.

A parameterized problem is fixed-parameter tractable (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ solves the problem in time bounded by $f(k) \cdot |I|^c$.

Parameterized reduction: given an input $I = (x, k)$ of the source problem, computes in time $f(k) \cdot |I|^c$, an equivalent instance $I' = (x', k')$ of the target problem, such that $k' \leq g(k)$ for some function g.

W[1]-hard problems: any problem that admits a parameterized reduction from Independent Set parameterized by the size of the solution.

A few words on parameterized complexity

Instances of a parameterized problem: come with an integer parameter k.

A parameterized problem is fixed-parameter tractable (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ solves the problem in time bounded by $f(k) \cdot |I|^c$.

Parameterized reduction: given an input $I = (x, k)$ of the source problem, computes in time $f(k) \cdot |I|^c$, an equivalent instance $I' = (x', k')$ of the target problem, such that $k' \leq g(k)$ for some function g.

W[1]-hard problems: any problem that admits a parameterized reduction from Independent Set parameterized by the size of the solution.

W[2]-hard problems: any problem that admits a parameterized reduction from Dominating Set parameterized by the size of the solution.

A few words on parameterized complexity

Instances of a parameterized problem: come with an integer parameter k.

A parameterized problem is fixed-parameter tractable (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ solves the problem in time bounded by $f(k) \cdot |I|^c$.

Parameterized reduction: given an input $I = (x, k)$ of the source problem, computes in time $f(k) \cdot |I|^c$, an equivalent instance $I' = (x', k')$ of the target problem, such that $k' \leq g(k)$ for some function g.

W[1]-hard problems: any problem that admits a parameterized reduction from Independent Set parameterized by the size of the solution.

W[2]-hard problems: any problem that admits a parameterized reduction from Dominating Set parameterized by the size of the solution.

The theory of parameterized complexity is built based on FPT $\neq$ W[1].

W[1]-hardness: strong evidence of not being FPT.
W[2]-hardness: even more!

Next section is...

Theorem (Araújo, Baste, S.)

Given a weighted forest (G, w), computing $\sigma(G, w)$ is W[1]*-hard parameterized by the size of a largest connected component of G.*

Theorem (Araújo, Baste, S.)

Given a weighted forest (G, w), computing $\sigma(G, w)$ is W[1]*-hard parameterized by the size of a largest connected component of G.*

Consequences: W[1]-hard parameterized by treewidth, cliquewidth, maximum degree, maximum diameter of a connected component, number of colors, etc.

Theorem (Araújo, Baste, S.)

Given a weighted forest (G, w), computing $\sigma(G, w)$ is W[1]-hard parameterized by the size of a largest connected component of G.

Consequences: W[1]-hard parameterized by treewidth, cliquewidth, maximum degree, maximum diameter of a connected component, number of colors, etc.

Theorem (Araújo, Baste, S.)

Given a weighted tree (G, w) and an integer r, computing $\sigma(G, w; r)$ is W[2]-hard parameterized by r.

Theorem (Araújo, Baste, S.)

Given a weighted forest (G, w), computing $\sigma(G, w)$ is W[1]-hard parameterized by the size of a largest connected component of G.

Consequences: W[1]-hard parameterized by treewidth, cliquewidth, maximum degree, maximum diameter of a connected component, number of colors, etc.

Theorem (Araújo, Baste, S.)

Given a weighted tree (G, w) and an integer r, computing $\sigma(G, w; r)$ is W[2]-hard parameterized by r.

Note: results are incomparable to those of [Araújo, Nisse, Pérennes. 2014]

Theorem (Araújo, Baste, S.)

Given a weighted forest (G, w), computing $\sigma(G, w)$ is W[1]-hard parameterized by the size of a largest connected component of G.

Consequences: W[1]-hard parameterized by treewidth, cliquewidth, maximum degree, maximum diameter of a connected component, number of colors, etc.

Theorem (Araújo, Baste, S.)

Given a weighted tree (G, w) and an integer r, computing $\sigma(G, w; r)$ is W[2]-hard parameterized by r.

Note: results are incomparable to those of [Araújo, Nisse, Pérennes. 2014]

Recall: on forests, $\sigma(G, w; r)$ can be computed in time $n^{O(r)}$.

Theorem (Araújo, Baste, S.)

Given a weighted forest (G, w), computing $\sigma(G, w)$ is W[1]-hard parameterized by the size of a largest connected component of G.

Consequences: W[1]-hard parameterized by treewidth, cliquewidth, maximum degree, maximum diameter of a connected component, number of colors, etc.

Theorem (Araújo, Baste, S.)

Given a weighted tree (G, w) and an integer r, computing $\sigma(G, w; r)$ is W[2]-hard parameterized by r.

Note: results are incomparable to those of [Araújo, Nisse, Pérennes. 2014]

Recall: on forests, $\sigma(G, w; r)$ can be computed in time $n^{O(r)}$.

Corollary (Araújo, Baste, S.)

Assuming ETH, there is no algorithm that, given a weighted tree (G, w) and a positive integer r, computes $\sigma(G, w; r)$ in time $f(r) \cdot n^{o(r)}$ for any computable function f.

Next section is...

General framework

Our reductions are inspired by the one of [Araújo, Nisse, Pérennes. 2014]

General framework

Our reductions are inspired by the one of [Araújo, Nisse, Pérennes. 2014]

We present two parameterized reductions:

1. Instance (G, k) of INDEPENDENT SET $\longrightarrow$
   Instance (G', w) of WEIGHTED COLORING.

General framework

Our reductions are inspired by the one of [Araújo, Nisse, Pérennes. 2014]

We present two parameterized reductions:

1. Instance (G, k) of INDEPENDENT SET $\longrightarrow$ Instance (G', w) of WEIGHTED COLORING.
    - There exists a solution of INDEPENDENT SET on (G, k) $\iff$ $\sigma(G', w) \leq M$, for some appropriately chosen real number $M < 2$.
    - The size of any connected component of G' is at most $13 \cdot 2^{4k} + 12$.

General framework

Our reductions are inspired by the one of [Araújo, Nisse, Pérennes. 2014]

We present two parameterized reductions:

1. Instance (G, k) of Independent Set $\longrightarrow$
   Instance (G', w) of Weighted Coloring.

   - There exists a solution of Independent Set on (G, k) $\iff$ $\sigma(G', w) \leq M$, for some appropriately chosen real number $M < 2$.
   - The size of any connected component of G' is at most $13 \cdot 2^{4k} + 12$.

2. Instance (G, k) of Dominating Set $\longrightarrow$
   Instance (G', w) of Weighted Coloring.

General framework

Our reductions are inspired by the one of [Araújo, Nisse, Pérennes. 2014]

We present two parameterized reductions:

1. Instance (G, k) of INDEPENDENT SET $\longrightarrow$ Instance (G', w) of WEIGHTED COLORING.
   - There exists a solution of INDEPENDENT SET on (G, k) $\iff$ $\sigma(G', w) \leq M$, for some appropriately chosen real number $M < 2$.
   - The size of any connected component of G' is at most $13 \cdot 2^{4k} + 12$.

2. Instance (G, k) of DOMINATING SET $\longrightarrow$ Instance (G', w) of WEIGHTED COLORING.
   - There exists a solution of DOMINATING SET on (G, k) $\iff$ $\sigma(G', w; r) \leq M$, with $r = 4k + 4$.

Some useful gadgets

For $i \in [0, 4k+3]$ and $j \in [0, n]$, let $w_i^j = \frac{1}{2^i} + j\varepsilon$, for some $\varepsilon > 0$.

Index i: colors in G'. Index j: vertices of the input graph G.

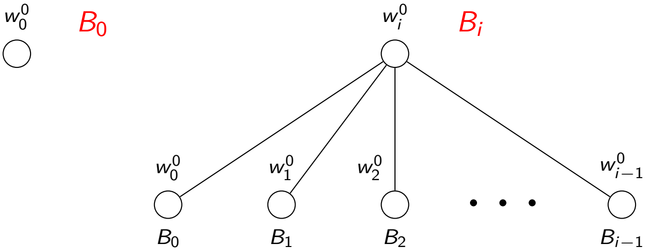
Some useful gadgets

For $i \in [0, 4k+3]$ and $j \in [0, n]$, let $w_i^j = \frac{1}{2^i} + j\varepsilon$, for some $\varepsilon > 0$.

Index i: colors in G'. Index j: vertices of the input graph G.

Binomial trees **Role**: force most of the colors of the vertices of the forest.

For each $i \in [0, 4k+3]$, we define recursively the weighted rooted tree B_i:



- if $i = 0$, then B_0 has a unique node of weight w_0^0,
- otherwise, B_i has a root r of weight w_i^0 and, for each $j \in [0, i-1]$, we introduce a copy of B_j and we connect its root to r.

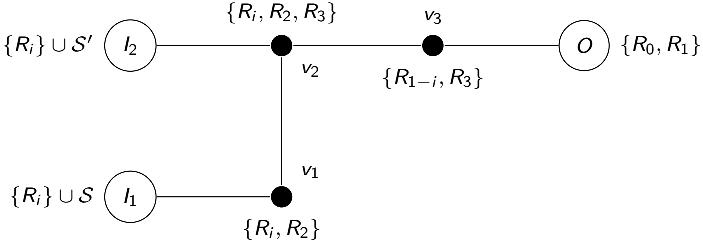
Some useful gadgets (2)

For $\ell \in [0, 3]$, let $W_\ell = w^0_{4k+\ell} = \frac{1}{2^{4k+\ell}}$.
Let $R_\ell = S_{4k+\ell}$ to be the unique color of weight W_ℓ.

For $\ell \in [0,3]$, let $W_\ell = w^0_{4k+\ell} = \frac{1}{2^{4k+\ell}}$.
Let $R_\ell = S_{4k+\ell}$ to be the unique color of weight W_ℓ.

AND gadget

Let $i \in [0,1]$. Given two vertices I_1, I_2, we define the R_i-AND gadget between the input vertices I_1 and I_2, to be "this" graph:



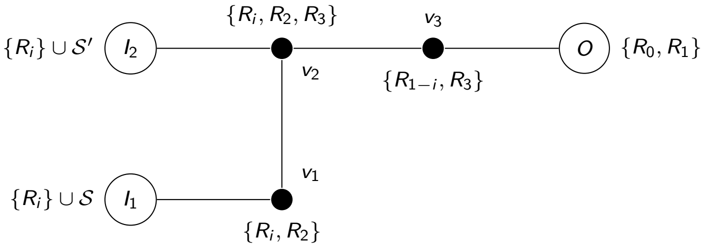
Available colors are forced by pendant binomial trees (omitted).

Some useful gadgets (2)

For $\ell \in [0,3]$, let $W_\ell = w^0_{4k+\ell} = \frac{1}{2^{4k+\ell}}$.
Let $R_\ell = S_{4k+\ell}$ to be the unique color of weight W_ℓ.

AND gadget

Let $i \in [0,1]$. Given two vertices I_1, I_2, we define the R_i-AND gadget between the input vertices I_1 and I_2, to be "this" graph:



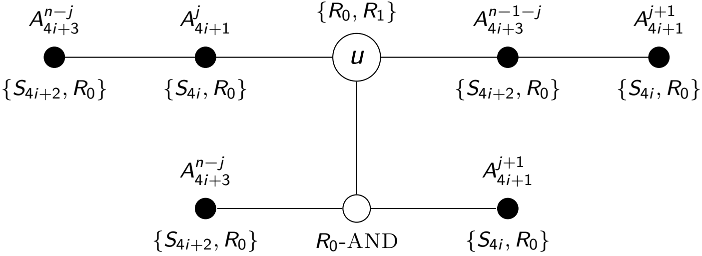
Available colors are forced by pendant binomial trees (omitted).

If both I_1 and I_2 are colored R_i, then O must be colored R_i.
If either I_1 or I_2 is not colored R_i, then O can be colored either R_0 or R_1.

Some useful gadgets (3)

Vertex tree

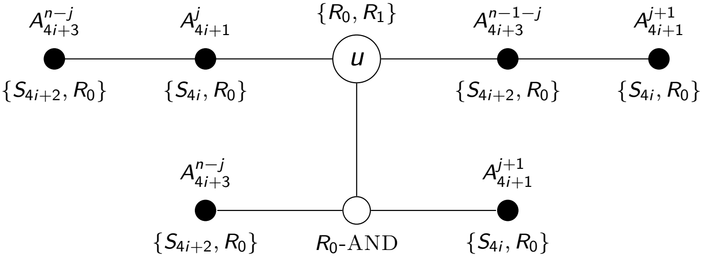
For $i \in [0, k-1]$ and $j \in [0, n-1]$, we define the vertex tree T_i^j, representing the vertex j, to be "this" graph, with root u:



Some useful gadgets (3)

Vertex tree

For $i \in [0, k-1]$ and $j \in [0, n-1]$, we define the vertex tree T_i^j, representing the vertex j, to be "this" graph, with root u:

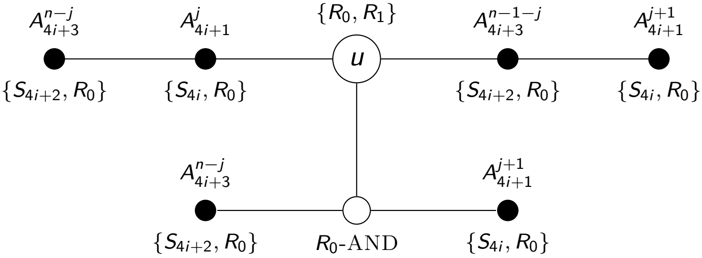


Idea: root u gets color R_0 (R_1) $\Rightarrow$ vertex v is (not) in the solution.
(It can be proved that the choices need to be consistent for each vertex.)

Some useful gadgets (3)

Vertex tree

For $i \in [0, k-1]$ and $j \in [0, n-1]$, we define the vertex tree T_i^j, representing the vertex j, to be "this" graph, with root u:



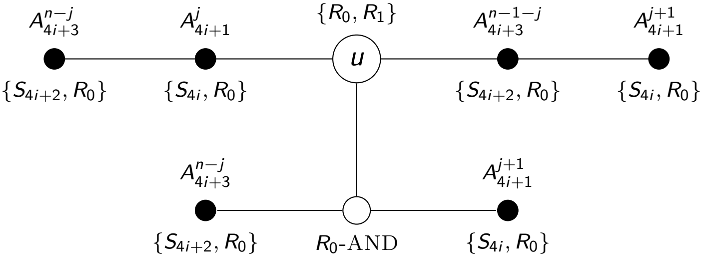
Idea: root u gets color R_0 (R_1) $\Rightarrow$ vertex v is (not) in the solution.
(It can be proved that the choices need to be consistent for each vertex.)

Each time we choose a vertex $\Rightarrow$ "pay" $(n-1)\varepsilon$ in the total weight.

Some useful gadgets (3)

Vertex tree

For $i \in [0, k-1]$ and $j \in [0, n-1]$, we define the vertex tree T_i^j, representing the vertex j, to be "this" graph, with root u:



Idea: root u gets color R_0 (R_1) $\Rightarrow$ vertex v is (not) in the solution.
(It can be proved that the choices need to be consistent for each vertex.)

Each time we choose a vertex $\Rightarrow$ "pay" $(n-1)\varepsilon$ in the total weight.

Making k such choices is forced by $M = k(n-1)\varepsilon + \sum_{i \in [0,4k+3]} \frac{1}{2^i}$.

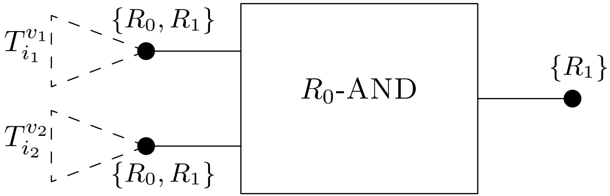
Sketch of the W[1]-hardness reduction

(G, k) of INDEPENDENT SET $\longrightarrow$ (G', w) of WEIGHTED COLORING.

Sketch of the W[1]-hardness reduction

(G, k) of INDEPENDENT SET $\longrightarrow$ (G', w) of WEIGHTED COLORING.

We create, for each edge $\{v_1, v_2\} \in E(G)$ and $i_1, i_2 \in [0, k-1]$, a tree $H_{\{v_1,v_2\},i_1,i_2}$ obtained from the vertex trees $T_{i_1}^{v_1}$ and $T_{i_2}^{v_2}$ as follows:

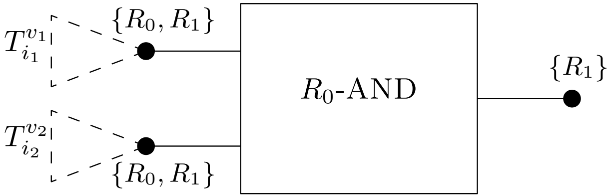


Idea: for $\{v_1, v_2\} \in E(G)$, at most one of v_1 and v_2 in the stable set.

Sketch of the W[1]-hardness reduction

(G, k) of INDEPENDENT SET $\longrightarrow$ (G', w) of WEIGHTED COLORING.

We create, for each edge $\{v_1, v_2\} \in E(G)$ and $i_1, i_2 \in [0, k-1]$, a tree $H_{\{v_1,v_2\},i_1,i_2}$ obtained from the vertex trees $T^{v_1}_{i_1}$ and $T^{v_2}_{i_2}$ as follows:



Idea: for $\{v_1, v_2\} \in E(G)$, at most one of v_1 and v_2 in the stable set.

Forest (G', w): disjoint union of these trees $H_{\{v_1,v_2\},i_1,i_2}$, for $\{v_1, v_2\} \in E(G)$ and $i_1, i_2 \in [0, k-1]$. (with some other technical stuff)

There exists a solution of INDEPENDENT SET on $(G, k) \Leftrightarrow \sigma(G', w) \leq M$.

Gràcies!