Weighted Coloring in Trees

Julio Araujo, Nicolas Nisse and Stéphane Pérennes

COATI, Inria, I3S (CNRS, Univ. Nice-Sophia Antipolis)

COATIs' Seminar

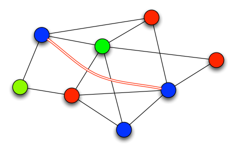
Sophia Antipolis, March 5^{th} 2013

Proper Coloring

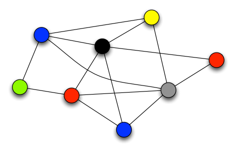
Let $G = (V, E)$ be a graph.

Proper Coloring:

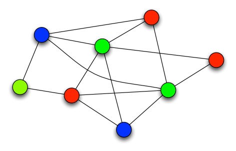
k-coloring:	$c : V \rightarrow \{1, \cdots, k\}$	Assign a color to each node
proper:	$\forall uv \in E,\ c(u) \neq c(v)$	adjacent nodes $\Rightarrow$ distinct colors



Unproper 3-coloring (red edge)



Proper 6-coloring



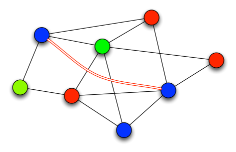
Proper 3-coloring

Proper Coloring

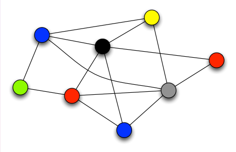
Let $G = (V, E)$ be a graph.

Proper Coloring:

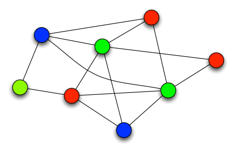
k-coloring:	$c : V \to \{1, \cdots, k\}$	Assign a color to each node
proper:	$\forall uv \in E,\ c(u) \neq c(v)$	adjacent nodes $\Rightarrow$ distinct colors



Unproper 3-coloring (red edge)



Proper 6-coloring



Proper 3-coloring

Chromatic number $\chi(G)$ — min. $k \in \mathbb{N}$ such that G has a k-coloring.

Minimum number of colors needed to proper color G
Computing χ: NP-complete in general graphs. Trivial in bipartite graphs.

Examples of applications: Four Colors' Theorem, shared resources allocations (e.g., frequency assignment, offices assignment in COATI, *etc.*)

Weighted Coloring

Let $G = (V, E)$ be a graph.
Weight function over vertices: $w : V \to \mathbb{R}^+$

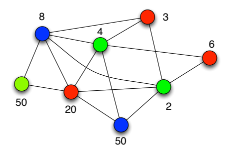
Weighted Coloring:

proper coloring: $c : V \to \{1, \cdots, k\}$ such that, $\forall uv \in E$, $c(u) \neq c(v)$

weight of a color $i \leq k$: $w(i) = \max\{w(v) \mid v \in V \text{ and } c(v) = i\}$

maximum weight of a node with color $i \in \{1, \cdots, k\}$

weight of coloring c: $w(c) = \sum_{i=1}^{k} w(i)$ — sum of weights of all colors



3-*coloring with weight*
$50 + 50 + 20 = 120$

Weighted Coloring

Let $G = (V, E)$ be a graph.
Weight function over vertices: $w : V \to \mathbb{R}^+$

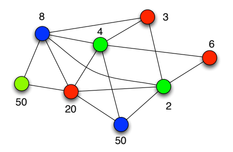
Weighted Coloring:

proper coloring: $c : V \to \{1, \cdots, k\}$ such that, $\forall uv \in E$, $c(u) \neq c(v)$

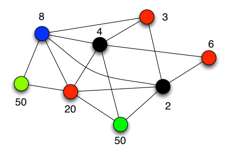
weight of a color $i \leq k$: $w(i) = \max\{w(v) \mid v \in V \text{ and } c(v) = i\}$

maximum weight of a node with color $i \in \{1, \cdots, k\}$

weight of coloring c: $w(c) = \sum_{i=1}^{k} w(i)$ sum of weights of all colors



3-coloring with weight
$50 + 50 + 20 = 120$



4-coloring with weight
$50 + 20 + 8 + 4 = 82$
Adding a color may decrease weight

Weighted Coloring

Let $G = (V, E)$ be a graph.
Weight function over vertices: $w : V \to \mathbb{R}^+$

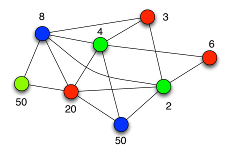
Weighted Coloring:

proper coloring: $c : V \to \{1, \cdots, k\}$ such that, $\forall uv \in E$, $c(u) \neq c(v)$

weight of a color $i \leq k$: $w(i) = \max\{w(v) \mid v \in V \text{ and } c(v) = i\}$

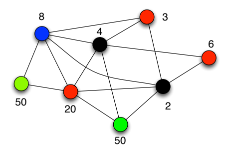
maximum weight of a node with color $i \in \{1, \cdots, k\}$

weight of coloring c: $w(c) = \sum_{i=1}^{k} w(i)$ sum of weights of all colors



3-coloring with weight

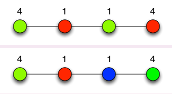
$50 + 50 + 20 = 120$



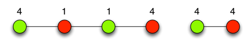
4-coloring with weight

$50 + 20 + 8 + 4 = 82$

Adding a color may decrease weight



Also in trees, using more colors may decrease the weight



Disjoint union behaves badly

Weighted Coloring

Let $G = (V, E)$ be a graph.
Weight function over vertices: $w : V \to \mathbb{R}^+$

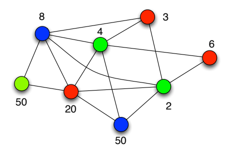
Weighted Coloring:

proper coloring: $c : V \to \{1, \cdots, k\}$ such that, $\forall uv \in E$, $c(u) \neq c(v)$

weight of a color $i \leq k$: $w(i) = \max\{w(v) \mid v \in V \text{ and } c(v) = i\}$

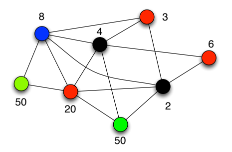
maximum weight of a node with color $i \in \{1, \cdots, k\}$

weight of coloring c: $w(c) = \sum_{i=1}^{k} w(i)$ sum of weights of all colors



3-coloring with weight

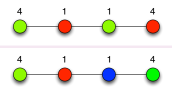
$50 + 50 + 20 = 120$



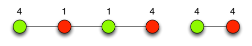
4-coloring with weight

$50 + 20 + 8 + 4 = 82$

Adding a color may decrease weight



Also in trees, using more colors may decrease the weight



Disjoint union behaves badly

Remark: If all nodes have weight 1 $\Rightarrow$ **weight of coloring = number of colors**

Weighted Coloring Problem [Guan & Zhu, 1997]

input: weighted graph (G, w), $p \in \mathbb{R}^+$
output: $\exists$ proper coloring of G with weight $\leq p$?

NP-complete

- general graphs (extends proper coloring)
- split/interval/triangle-free planar/bipartite graphs [DWMP 02, EMP 06, WDEMP 09]

Remark: the number of colors is not fixed

It always exists optimal coloring using $\leq \Gamma$ (*grundy number*) colors [Guan & Zhu 97]

In bounded treewidth (tw) n-node graphs

- if the number r of colors is fixed, exact algorithm in $n^{O(r)}$ [Guan & Zhu 97]
- grundy number $\Gamma = O(tw \cdot \log n)$ [Linhares & Reed 06]
- polynomial-time approximation scheme (PTAS) [Escoffier *et al.* 06]

Weighted Coloring Problem in Trees

Open questions in [Guan & Zhu, 1997]

Complexity of computing, in trees (T, w),

- the minimum weight $\chi_w(T)$ of a proper coloring? (Weighted Coloring Problem)
- a proper coloring with minimum weight?
- the minimum number of colors used by a proper coloring with minimum weight?

Partial answers

- Algorithm to compute optimal coloring in time $n^{O(\log n)}$

  straightforward from results of previous slide because trees have $tw = 1$
- in P when nodes of degree ≥ 3 form an independent set [Kavitha & Mestre 12]

Weighted Coloring Problem in Trees

Open questions in [Guan & Zhu, 1997]

Complexity of computing, in trees (T, w),

- the minimum weight $\chi_w(T)$ of a proper coloring? (Weighted Coloring Problem)
- a proper coloring with minimum weight?
- the minimum number of colors used by a proper coloring with minimum weight?

Partial answers

- Algorithm to compute optimal coloring in time $n^{O(\log n)}$

  straightforward from results of previous slide because trees have $tw = 1$
- in P when nodes of degree ≥ 3 form an independent set [Kavitha & Mestre 12]

Theorem 1 [Araujo, Nisse, Pérennes]

If 3-SAT cannot be solved in sub-exponential time (ETH), then
Weighted Coloring Problem cannot be solved in trees in time less than $n^{\Omega(\log n)}$

with n the size of the input tree.

3-SAT and NP-hierarchy

3-SAT Problem $\Phi(v_1, \cdots, v_\eta)$ boolean formula in 3-Conjunctive Normal Form.
$\exists$ truth-assignment that satisfies Φ?

$\Phi(a, b, c, d, e) = (a \vee \bar{b} \vee c) \wedge (\bar{e} \vee b \vee \bar{c}) \wedge (\bar{a} \vee \bar{c} \vee e)$

$\Phi(a) = (a \vee a \vee a) \wedge (\bar{a} \vee \bar{a} \vee \bar{a})$

3-SAT and NP-hierarchy

3-SAT Problem $\Phi(v_1, \cdots, v_\eta)$ boolean formula in 3-Conjunctive Normal Form.
$\exists$ truth-assignment that satisfies Φ?

$\Phi(a,b,c,d,e) = (a \vee \bar{b} \vee c) \wedge (\bar{e} \vee b \vee \bar{c}) \wedge (\bar{a} \vee \bar{c} \vee e)$ $\Phi(1,1,0,0,0) =$ TRUE

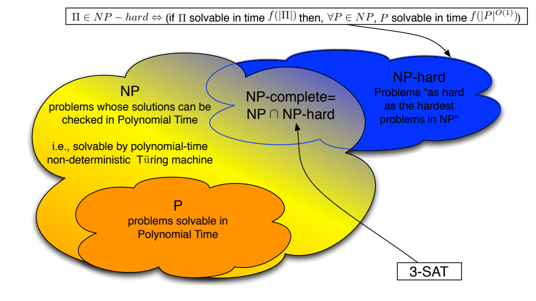
$\Phi(a) = (a \vee a \vee a) \wedge (\bar{a} \vee \bar{a} \vee \bar{a})$ $\Phi(1) = \Phi(0) =$ FALSE, then NOT SATISFIABLE

3-SAT and NP-hierarchy

3-SAT Problem $\Phi(v_1, \cdots, v_\eta)$ boolean formula in 3-Conjunctive Normal Form.
$\exists$ truth-assignment that satisfies Φ?

$\Phi(a,b,c,d,e) = (a \vee \bar{b} \vee c) \wedge (\bar{e} \vee b \vee \bar{c}) \wedge (\bar{a} \vee \bar{c} \vee e)$ $\Phi(1,1,0,0,0) = \text{TRUE}$

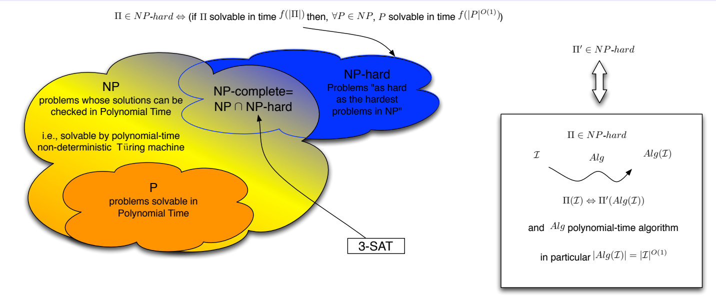
$\Phi(a) = (a \vee a \vee a) \wedge (\bar{a} \vee \bar{a} \vee \bar{a})$ $\Phi(1) = \Phi(0) = \text{FALSE}$, then NOT SATISFIABLE



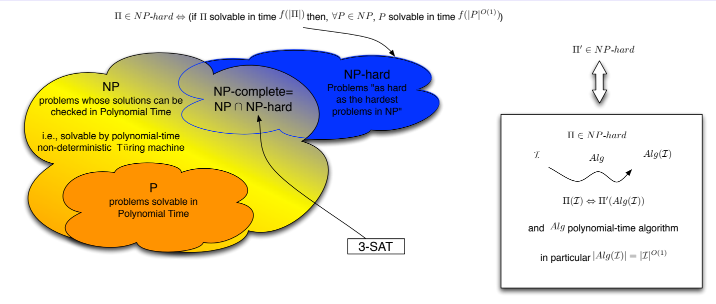
Cook's Theorem 3-SAT is "the" NP-complete problem

$\Rightarrow$ no polynomial-time ($\eta^{O(1)}$) algorithm to solve 3-SAT unless $P = NP$

3-SAT, ETH and reductions

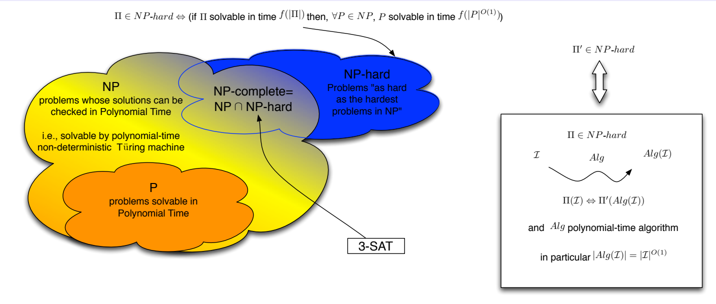


3-SAT, ETH and reductions



Conjecture: Exponential Time Hypothesis (ETH) $(\Rightarrow P \neq NP)$

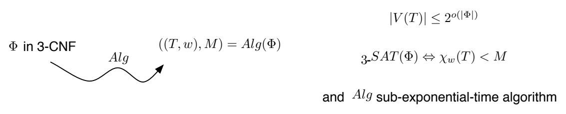
There is no sub-exponential-time ($2^{o(\eta)}$) algorithm to solve 3-SAT



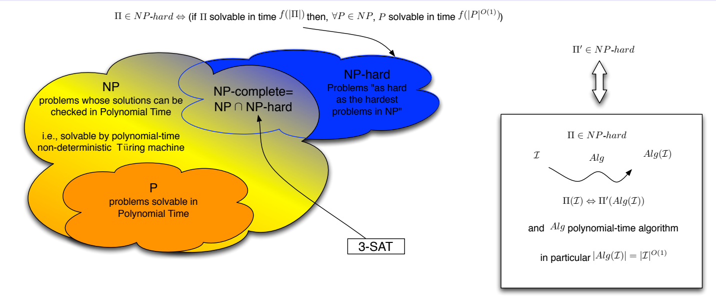
Conjecture: Exponential Time Hypothesis (ETH) $(\Rightarrow P \neq NP)$

There is no sub-exponential-time ($2^{o(\eta)}$) algorithm to solve 3-SAT

Proof of Th.1: $\exists Alg$ s.t.



3-SAT, ETH and reductions



Conjecture: Exponential Time Hypothesis (ETH) ($\Rightarrow P \neq NP$)

There is no sub-exponential-time ($2^{o(\eta)}$) algorithm to solve 3-SAT

We give a $2^{O(\sqrt{\eta})}$-time algorithm that:

INPUT: any boolean formula $\Phi(v_1, \cdots, v_\eta)$ in 3-CNF (# of clauses $\eta^{O(1)}$)
OUTPUT: a weighted tree (T, w) such that

$$|V(T)| = 2^{O(\sqrt{\eta})}, \texttt{ w encodable with } 2^{O(\sqrt{\eta})} \texttt{ bits,}$$

$$\chi_w(T) < M \texttt{ if and only if } \Phi \in SAT.$$

3-SAT to INT-SAT to Weighted Coloring

We give a $2^{O(\sqrt{\eta})}$-time algorithm that:

INPUT: any boolean formula $\Phi(v_1, \cdots, v_\eta)$ in 3-CNF (# of clauses $\eta^{O(1)}$)
OUTPUT: a weighted tree (T, w) such that

$$|V(T)| = 2^{O(\sqrt{\eta})}, \texttt{ w encodable with } 2^{O(\sqrt{\eta})} \texttt{ bits,}$$

$$\chi_w(T) < M \texttt{ if and only if } \Phi \in SAT.$$

3-SAT to INT-SAT to Weighted Coloring

We give a $2^{O(\sqrt{\eta})}$-time algorithm that:

INPUT: any boolean formula $\Phi(v_1, \cdots, v_\eta)$ in 3-CNF (# of clauses $\eta^{O(1)}$)
OUTPUT: a weighted tree (T, w) such that

$$|V(T)| = 2^{O(\sqrt{\eta})}, \texttt{ w encodable with } 2^{O(\sqrt{\eta})} \texttt{ bits,}$$

$$\chi_w(T) < M \texttt{ if and only if } \Phi \in SAT.$$

what Stéphane would like: each variable corresponds to the choice of one color
but: size of the tree is exponential in the # of colors :(
what Stéphane did: from η boolean variables to $\sqrt{\eta}$ variables in $\{0, \cdots, 2^{\sqrt{\eta}}\text{-}1\}$

η variables	$v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}$			
$\sqrt{\eta} \times \sqrt{\eta}$ booleans	0 1 0 0	1 1 1 0	0 1 0 1	0 0 0 1
$\sqrt{\eta}$ integers in $\{0, \cdots, 2^{\sqrt{\eta}} - 1\}$	4	14	5	1
$\sqrt{\eta} \cdot 2^{\sqrt{\eta}}$ booleans	0000000000010000	0100000000000000	0000000000100000	0000000000000010

3-SAT to INT-SAT to Weighted Coloring

We give a $2^{O(\sqrt{\eta})}$-time algorithm that:

INPUT: any boolean formula $\Phi(v_1, \cdots, v_\eta)$ in 3-CNF (# of clauses $\eta^{O(1)}$)
OUTPUT: a weighted tree (T, w) such that

$$|V(T)| = 2^{O(\sqrt{\eta})}, \texttt{ w encodable with } 2^{O(\sqrt{\eta})} \texttt{ bits},$$

$$\chi_w(T) < M \texttt{ if and only if } \Phi \in SAT.$$

Algorithm 1: $2^{O(\sqrt{\eta})}$-time

INPUT: any boolean formula $\Phi(v_1, \cdots, v_\eta)$ in 3-CNF (# of clauses $\eta^{O(1)}$)
OUTPUT: boolean formula $\Phi_{int}(y_1^1, \cdots, y_{2^{\sqrt{\eta}}}^1, \cdots, y_1^{\sqrt{\eta}}, \cdots, y_{2^{\sqrt{\eta}}}^{\sqrt{\eta}})$ in CNF

$\Phi \in SAT \quad \Leftrightarrow \quad \exists$ truth assignment of Φ_{int} s.t. $\forall j \leq \sqrt{\eta}$ there is at most one $i \leq 2^{\sqrt{\eta}}$ s.t. $y_i^j = 1$

$\Leftrightarrow \quad \exists$ truth assignment of Φ_{int} corresponding to $(j_1, \cdots, j_{\sqrt{\eta}}) \in \{0, \cdots, 2^{\sqrt{\eta}}\text{-}1\}^{\sqrt{\eta}}$.

Algorithm 2: $2^{O(\sqrt{\eta})}$-time

INPUT: any boolean formula $\Phi_{int}(y_1^1, \cdots, y_n^1, \cdots, y_1^m, \cdots, y_n^m)$ in CNF,
OUTPUT: a weighted tree (T, w) s.t. $|V(T)| = poly(nm)2^{O(m)}$

$\chi_w(T) < M \quad \Leftrightarrow \quad \exists$ integral assignment of Φ_{int} (i.e., $\forall j \leq m$ there is at most one $i \leq n$ s.t. $y_i^j = 1$).

Building the tree T

Boolean formula $\Phi_{int}(y_1^1, \cdots, y_n^1, \cdots, y_1^m, \cdots, y_n^m)$ in CNF
integral-assignments s.t. $\forall i \leq m$ there is at most one $j \leq n$ s.t. $y_j^i = 1$

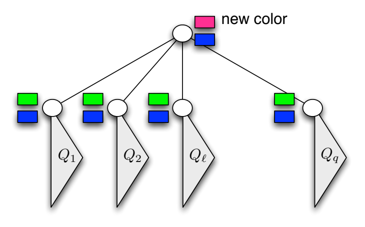
Clause tree — truth $\Leftrightarrow$ choice in coloring

One subtree Q_ℓ per clause c_ℓ.
Kind of correspondence between integral assignments and colorings with weight $\leq M$.

$c_\ell = 1 \Leftrightarrow$ THE ROOT OF Q_ℓ CAN TAKE COLOR Δ_1 OR Δ

$c_\ell = 0 \Leftrightarrow$ THE ROOT OF Q_ℓ MUST HAVE COLOR Δ_1

The tree T — (squares indicate possible colors)



If one clause is wrong $\Rightarrow$ colored Δ_1 $\Rightarrow$ the root of T colored C $\Rightarrow$ weight $> M$.

Building the tree T

Boolean formula $\Phi_{int}(y_1^1, \cdots, y_n^1, \cdots, y_1^m, \cdots, y_n^m)$ in CNF

integral-assignments s.t. $\forall i \leq m$ there is at most one $j \leq n$ s.t. $y_j^i = 1$

Variable trees

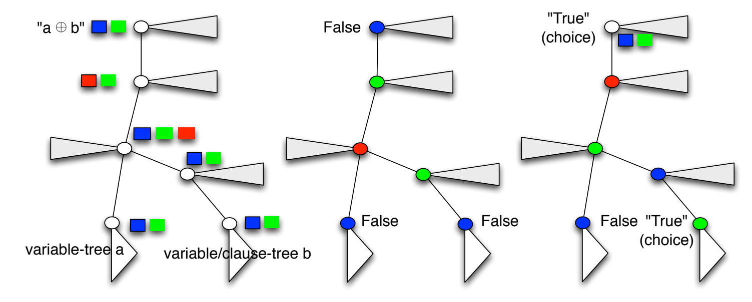
One subtree T_j^i per occurrence of variable y_j^i.

Kind of correspondence between integral assignments and colorings with weight $\leq M$.

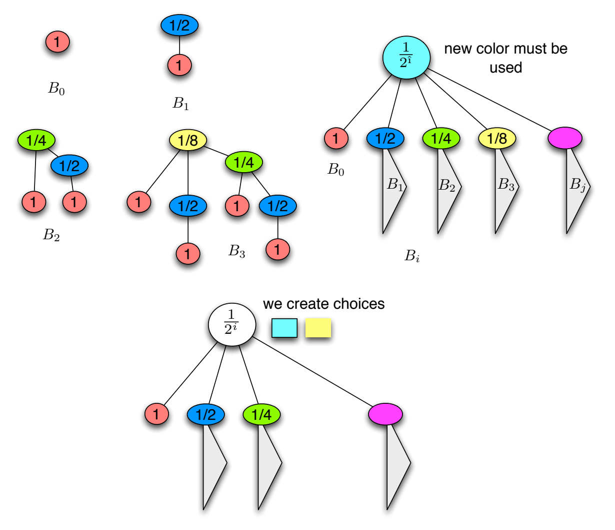
$y_i^j = 1 \Leftrightarrow$ THE ROOT OF T_i^j CAN TAKE COLOR Δ_1 OR Δ

$y_i^j = 0 \Leftrightarrow$ THE ROOT OF T_i^j MUST HAVE COLOR Δ_1

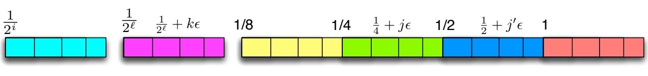
To make an "OR" and the Clause trees (squares indicate possible colors)



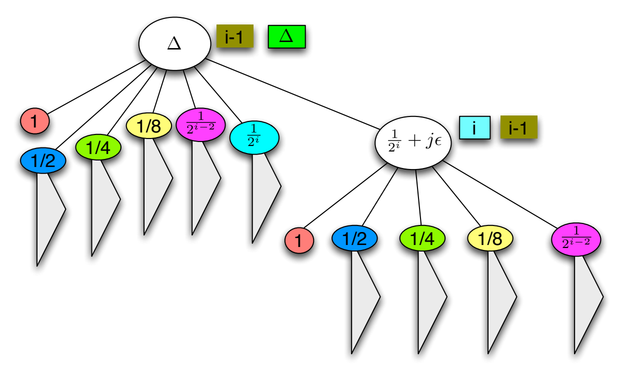
Force and forbid some colors



Hierarchy of colors and choices in their value



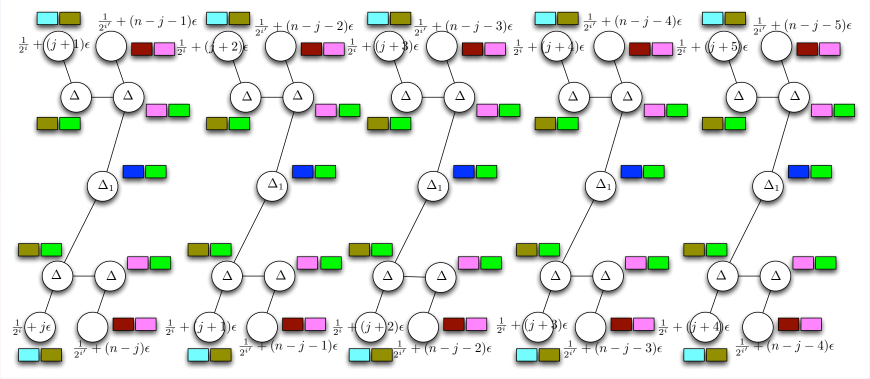
Each color i has weight $\frac{1}{2^i} + j_i\epsilon$ for some $j_i \in \{1 \cdots n\}$.



Either we "pay" $j\epsilon$ for color i, or the root must receive Color Δ.

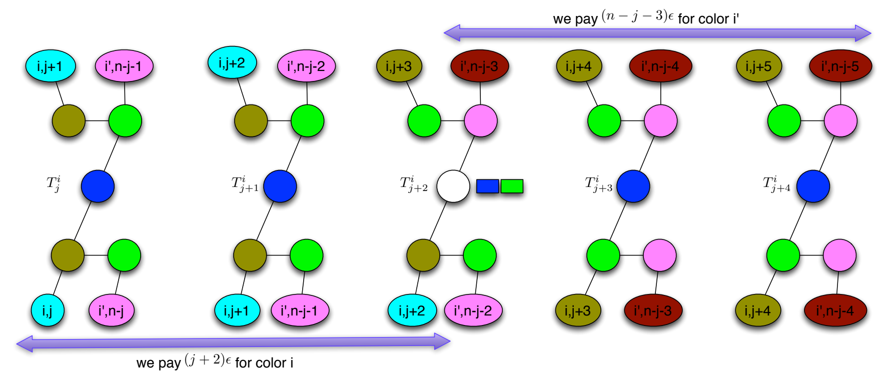
The final touch

This picture is impossible to understand, but I spent some time to do it :P



The final touch

Each coloring with weight $\leq M$ has particular structure: corresponds to choose j_i for each color i.



Once j_i is chosen, for each i, we have only a choice for the root of $T^i_{j_i}$.

Further work

- Is the weighted coloring problem NP-complete in trees?
- Can our technique, using integral version of SAT, be used to prove other complexity results?
- Julio would like to study the complexity of weighted coloring in P_{127}-almost-sparse-mais-pas-trop-graphs.