



## Weighted Coloring in Trees

Julio Araujo, Nicolas Nisse, Stéphane Pérennes

### ► To cite this version:

Julio Araujo, Nicolas Nisse, Stéphane Pérennes. Weighted Coloring in Trees. [Research Report] RR-8249, INRIA. 2013. hal-00794622v2

**HAL Id: hal-00794622**

**<https://inria.hal.science/hal-00794622v2>**

Submitted on 7 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Weighted Coloring in Trees

Julio Araujo, Nicolas Nisse, Stéphane Pérennes

**RESEARCH  
REPORT**

**N° 8249**

February 2013

Project-Teams COATI





## Weighted Coloring in Trees\*

Julio Araujo, Nicolas Nisse, Stéphane Pérennes

Project-Teams COATI

Research Report n° 8249 — February 2013 — 17 pages

**Abstract:** A proper coloring of a graph is a partition of its vertex set into stable sets, where each part corresponds to a *color*. For a vertex-weighted graph, the *weight of a color* is the maximum weight of its vertices. The *weight of a coloring* is the sum of the weights of its colors. Guan and Zhu defined the *weighted chromatic number* of a vertex-weighted graph  $G$  as the smallest weight of a proper coloring of  $G$  (1997). If vertices of a graph have weight 1, its weighted chromatic number coincides with its chromatic number. Therefore, the problem of computing the weighted chromatic number is NP-complete in general graphs. This problem remains NP-complete in some particular graph classes as bipartite graphs.

In their seminal paper, Guan and Zhu asked whether the weighted chromatic number of bounded tree-width graphs (partial  $k$ -trees) can be computed in polynomial-time. Escoffier et al. designed a polynomial-time approximation scheme for computing the weighted chromatic number of partial  $k$ -trees (2006), and Kavitha and Mestre provided polynomial-time exact algorithms for sub-classes of trees (2009). Surprisingly, the time-complexity of computing this parameter in trees is still open. The Exponential Time Hypothesis (ETH) states that 3-SAT cannot be solved in sub-exponential time. We show that, assuming ETH, the best algorithm to compute the weighted chromatic number of  $n$ -node trees has time-complexity  $n^{\Theta(\log n)}$ . Our result mainly relies on proving that, when computing an optimal proper weighted coloring of a graph  $G$ , it is hard to combine colorings of its connected components, even when  $G$  is a forest.

**Key-words:** Weighted Coloring; Exponential Time Hypothesis; 3-SAT; Trees.

---

\* This work was partly funded by the ANR projects AGAPE and GRATEL, and promoted by the INRIA/FUNCAP project ALERTE and the INRIA associate-team AIDyNet. Due to lack of space, most of the proofs have been sketched. The full paper may be found in [1].

RESEARCH CENTRE  
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93  
06902 Sophia Antipolis Cedex

## Coloration pondérée dans les arbres

**Résumé :** Nous prouvons que, en supposant qu'il n'existe aucun algorithme sous-exponentiel pour résoudre 3-SAT (ETH), alors le meilleur algorithme pour résoudre le problème de coloration pondérée dans les arbres a une complexité de  $n^{\Theta(\log n)}$ , où  $n$  est la taille de l'entrée.

**Mots-clés :** Coloration pondérée; ETH; 3-SAT; arbre

## 1 Introduction

Given a loop-less graph  $G = (V, E)$ , a (*proper*)  $k$ -coloring of  $G$  is a surjective function  $c : V \rightarrow \{1, \dots, k\}$  that assigns to each vertex  $v \in V$  a *color*  $c(v) \in \{1, \dots, k\}$ , such that, for any  $\{u, v\} \in E$ ,  $c(u) \neq c(v)$ . Equivalently, a  $k$ -coloring of  $G$  is a partition  $c = (S_1, \dots, S_k)$  of  $V$  such that, for any  $1 \leq i \leq k$ ,  $S_i$  is a non-empty independent set of vertices that have the same color  $i$ . One of the most studied problems in Graph Theory consists in minimizing the number of colors of a proper coloring of a graph. Namely, GRAPH COLORING aims at computing the *chromatic number* of a graph  $G$ , denoted by  $\chi(G)$ , which is the minimum  $k$  for which  $G$  has a  $k$ -coloring. This is one of the Karp's NP-hard problems [7].

In [5], Guan and Zhu generalized GRAPH COLORING to vertex-weighted graphs. A (*vertex*) *weighted graph*  $(G, w)$  consists of a loop-less graph  $G = (V, E)$  and a weight function  $w : V \rightarrow \mathbb{R}_+$  over the vertices of  $G$ . Given a  $k$ -coloring  $c = (S_1, \dots, S_k)$  of a weighted graph  $(G, w)$ , the *weight of color*  $i$  ( $1 \leq i \leq k$ ) is defined by  $w(i) = \max_{v \in S_i} w(v)$ . The *weight of coloring*  $c$  is  $w(c) = \sum_{i=1}^k w(i)$ . The *weighted chromatic number* of  $(G, w)$ , denoted by  $\chi_w(G)$ , is the minimum weight of a proper coloring of  $(G, w)$ . The WEIGHTED COLORING Problem takes a weighted graph  $(G, w)$  as input and aims at computing  $\chi_w(G)$  [5].

Observe that if the weight of each of the vertices of a graph  $(G, w)$  is equal to one, then the weight of a coloring is the number of its colors and thus,  $\chi_w(G) = \chi(G)$ . Therefore, WEIGHTED COLORING generalizes GRAPH COLORING to weighted graphs, and, as a consequence, this problem is NP-hard in general graphs. Moreover, WEIGHTED COLORING has been shown NP-hard in bipartite graphs [3], where GRAPH COLORING is trivial. In the last years, the WEIGHTED COLORING Problem has been addressed several times, however the complexity of this problem is surprisingly still unknown in the class of trees.

Here, we show that, if 3-SAT cannot be solved in sub-exponential time (Exponential Time Hypothesis), then WEIGHTED COLORING in trees is not in P.

**Related work.** Guan and Zhu defined the WEIGHTED COLORING Problem to model various problems of shared resources allocation. For instance, many problems consider a set of processes that use some common resources (memory, medium of communication, etc.) and aim at minimizing the total amount of used resources. Indeed, it is possible to mutualize resources: given a subset of processes that do not simultaneously use the resources, they can be assigned the same resources. The amount of resources necessary for this subset of processes is given by the process that uses the maximum amount of resources. Such problems can clearly be modeled as WEIGHTED COLORING in a conflict graph where processes are the nodes, the weight of a node is the amount of resources required by the corresponding process, and two nodes are adjacent if and only if the corresponding processes use resources simultaneously. More precisely, Guan and Zhu studied practical applications such as the DYNAMIC STORAGE ALLOCATION PROBLEM and the Distributed Dual Bus Network Media Access Control Protocol, which is a standard IEEE802.6 for metropolitan networks [5].

WEIGHTED COLORING has been shown to be NP-hard in the classes of split graphs, interval graphs, triangle-free planar graphs with bounded degree, and bipartite graphs [3, 10, 4]. On the other hand, the weighted chromatic number of cographs and of some subclasses of bipartite graphs can be found in polynomial-time [3, 10]. In their seminal paper, Guan and Zhu showed that, given a fixed parameter  $r \in \mathbb{N}$ , the minimum weight of a coloring using at most  $r$  colors can be computed in polynomial-time<sup>1</sup> in the class of bounded treewidth graphs (a.k.a. partial  $k$ -trees) [5]. They let open the question of the time-complexity of the WEIGHTED COLORING Problem in this class (partial  $k$ -trees) and, in particular, in trees. Later on, Escoffier et al. proposed a polynomial-time approximation scheme to compute the weighted chromatic number of bounded treewidth

<sup>1</sup>We emphasize that this algorithm is exponential in  $r$

graphs [4]. Kavitha and Mestre recently presented polynomial-time algorithms for subclasses of trees [8]. More precisely, they show that computing the weighted chromatic number can be done in linear time in the class of trees where nodes with degree at least three induce a stable set [8].

In the last years, many studies have been done on the WEIGHTED COLORING Problem, however the complexity of this natural problem was still unknown on trees. Indeed, WEIGHTED COLORING in trees has some intriguing properties: on the one hand, a reduction to another NP-hard problem was unlikely to exist due to the existence of a sub-exponential algorithm that we present in Section 2; on the other hand, all the classical methods to derive polynomial-time algorithms on trees failed [4, 8]. We provide here some explanation for these facts.

**Our results.** We show that, under the Exponential Time Hypothesis (ETH) (see Section 2), the best algorithm to compute the weighted chromatic number of trees has time-complexity  $n^{\Theta(\log n)}$ , where  $n$  is the number of vertices of the input tree.

The existence of an algorithm that solves the WEIGHTED COLORING Problem in time  $n^{\Theta(\log n)}$  follows easily from previous results. The difficulty is to prove that it is optimal under ETH. For this purpose, we show that computing the weighted chromatic number of an  $n$ -node tree is as hard as deciding whether a 3-SAT formula with size  $\log^2 n$  can be satisfied. So, our reduction is rather complex, but we hope that it contains ideas that may be used in other contexts. Along the line of our reduction, one will discover another surprising aspect: the difficulty of the problem not only comes from the graph structure, but rather relies on the way weights are structured. This implies that choosing the right color for a node is hard. We indeed use non binary constraint satisfaction formulae as main tool. Lastly, our reduction also proves that computing an optimal weighted coloring of a disconnected graph may be hard even if optimal colorings of each of its components can be done in polynomial-time.

**Organization of the paper.** The remainder of the paper is organized as follows. In Section 2, we formally state the main results of the paper: in Section 2.1, an  $n^{\mathcal{O}(\log n)}$ -time algorithm is derived from previous works, and in Section 2.2 we prove our main result assuming a technical reduction (Proposition 2). The remaining part of the paper is devoted to the proof of Proposition 2. In Section 3, we give the main ideas of its proof. Finally, in Section 4, we prove the Proposition 3 which allows to prove Proposition 2.

## 2 Preliminaries

### 2.1 Sub-exponential algorithm

In this section, we show that there exists a sub-exponential algorithm to solve the WEIGHTED COLORING Problem in trees and more generally in the class of bounded treewidth graphs. Actually, this is an almost trivial consequence of previous works that mainly relies on the number of colors used by weighted colorings in this class of graphs.

It is easy to see that there exist weighted graphs  $G$  for which any optimal weighted coloring uses strictly more than  $\chi(G)$  colors. For instance, let us consider the 4-node path  $P_4$  with  $V(P_4) = \{a, b, c, d\}$ ,  $w(a) = w(d) = 4$  and  $w(b) = w(c) = 1$  (see Figure 1). Any coloring of  $P_4$  with  $2 = \chi(P_4)$  colors has weight 8, and the optimal coloring  $\{\{a, d\}, \{b\}, \{c\}\}$  of  $P_4$  has weight  $\chi_w(P_4) = 6$  but uses 3 colors.

Hopefully, the number of colors used by optimal weighted colorings can be bounded by  $O(\log n)$  in the class of bounded treewidth graphs with  $n$  nodes. Indeed, in their seminal paper, Guan and Zhu studied the number of colors used by an optimal weighted coloring [5]. More precisely, they proved that the maximum number of colors of an optimal weighted coloring of a weighted graph  $(G, w)$  is its first-fit chromatic number  $\chi_{FF}(G)$  (a.k.a., *Grundy number*) [5].

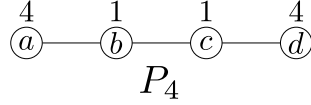


Figure 1: The (unique) optimal weighted coloring of the path  $P_4$  uses strictly more than  $\chi(P_4)$  colors.

Moreover, for any graph  $G$ , there exists a weight function  $w$  such that an optimal weighted coloring of  $(G, w)$  uses  $\chi_{FF}(G)$  colors. On the other hand, for any  $n$ -node graph  $G$  with tree-width at most  $k$ ,  $\chi_{FF}(G) = \mathcal{O}(k \log n)$  [9]. In particular, this implies that, for any  $n$ -node tree, there is an optimal weighted coloring using  $\mathcal{O}(\log n)$  colors. Finally, in the class of bounded treewidth graphs and when the number  $r \in \mathbb{N}$  of colors is fixed, there is an algorithm (using dynamic programming on the tree-decomposition) that computes the minimum weight of a coloring using at most  $r$  colors in time  $\mathcal{O}(n^r)$  where  $n$  is the number of vertices of the input graph [5].

By combining these results, the following proposition is straightforward:

**Proposition 1** *There exists an algorithm that solves the WEIGHTED COLORING Problem in time  $n^{\mathcal{O}(\log n)}$  in the class of bounded treewidth graphs, where  $n$  is the number of vertices of the input graph.*

## 2.2 Main Result

We now state formally our results that are subject to the Exponential Time Hypothesis.

Recall that an instance of the 3-SAT Problem is any Boolean formula  $\Phi(v_1, \dots, v_\eta)$  over the variables  $v_1, \dots, v_\eta$  in the conjunctive normal form (CNF) where each clause involves three variables. The size of  $\Phi$  is  $\eta$  if it depends on  $\eta$  variables and its number of clauses is polynomial in  $\eta$ . The 3-SAT Problem aims at deciding whether there exists a truth assignment to the variables such that  $\Phi(v_1, \dots, v_\eta)$  is true. It is well known that the 3-SAT Problem is NP-complete [2]. A fundamental question is to know whether it can be solved in sub-exponential time.

**Conjecture 1 (Exponential Time Hypothesis (ETH))** *3-SAT cannot be solved in time  $2^{o(\eta)}$  where  $\eta$  is the size of the instance [6].*

Observe that proving ETH would imply that  $P \neq NP$ .

The main part of this paper is devoted to prove the following result.

**Proposition 2** *For any Boolean formula  $\Phi$  of size  $\eta$ , there exist a weighted tree  $(T, w)$  with  $n = 2^{\mathcal{O}(\sqrt{\eta})}$  vertices and  $M \in \mathbb{R}$  such that  $\Phi$  is satisfiable if and only if  $\chi_w(T) < M$ . Moreover,  $(T, w)$  and  $M$  are computable in polynomial time in  $n$ , i.e., in time  $2^{o(\eta)}$ .*

Proposition 2 allow us to prove that there is no polynomial-time algorithm to solve the WEIGHTED COLORING Problem in trees, unless ETH fails.

**Theorem 2** *If ETH is true, then the best algorithm to compute the weighted chromatic number of a tree  $T$  has time-complexity  $n^{\Theta(\log n)}$  where  $n$  is the number of nodes of  $T$ .*

**Proof.** The existence of such an algorithm directly follows from Proposition 1.

For purpose of contradiction, let us assume that there exists an algorithm  $\mathcal{A}$  that solves the WEIGHTED COLORING Problem in time  $n^{\mathcal{O}(\log n)}$  in the class of trees, where  $n$  is the number of vertices of the input tree.

Let  $\Phi$  be any Boolean formula of size  $\eta$ . By Proposition 2, there exists a weighted tree  $(T, w)$  with  $n = 2^{\mathcal{O}(\sqrt{\eta})} = 2^{o(\eta)}$  vertices and  $M \in \mathbb{R}$  such that  $\Phi$  is satisfiable if and only if  $\chi_w(T) < M$ .



Consider the following algorithm to solve 3-SAT. For any Boolean formula  $\Phi$  of size  $\eta$ , first compute  $(T, w)$  and  $M$  in time  $2^{o(\eta)}$ , then use Algorithm  $\mathcal{A}$  to compute  $\chi_w(T)$  in time  $n^{o(\log n)} = 2^{o((\log n)^2)} = 2^{o(\eta)}$ . By definition,  $\Phi$  is satisfiable if and only if  $\chi_w(T) < M$ .

Therefore, the above algorithm solves the 3-SAT Problem in time  $2^{o(\eta)}$  where  $\eta$  is the size of the instance. This contradicts ETH. ■

The remaining part of the paper is devoted to the proof of Proposition 2.

### 3 Guideline of the proof of Proposition 2

Proposition 2 establishes a link between the WEIGHTED COLORING Problem and 3-SAT. Our proof is quite technical, so this section is devoted to describe the main difficulties and ingredients of the proof. We also try to give some intuitions.

#### 3.1 From boolean variables to integral variables

##### 3.1.1 WEIGHTED COLORING vs. SAT.

When computing a coloring of a weighted graph, we are facing choices of the following kind: to color a node  $v$ , either we use a color  $c$  that is already used without increasing its weight, i.e.,  $c$  has already been assigned to a vertex with weight greater than  $v$ , or the color  $c$  used for  $v$  increases the global weight, i.e., either  $c$  has not been used yet or the weight of  $v$  is greater than the weights of nodes previously colored with  $c$ . For instance, in the example of Figure 1, once the first two colors have been assigned to  $a$  and  $b$ , we have to decide either to assign a new color to  $c$  which increases the global weight, or to use again the color of  $a$ . Of course, there are other ways to proceed, but this example illustrates a choice that consists in either increasing the weight of one color or not doing it. In the example, we can choose to increase the weight of a new color from 0 to  $w(c)$  (by assigning this new color to  $c$ ) or not to do it; this latter choice would however lead to a worse coloring. Intuitively, for a boolean formula  $\Phi$ , we will build a tree such that the choices of the weights of the colors, in any coloring with small (depending on  $\Phi$ ) weight, will reflect the truth assignment of the variables of  $\Phi$ .

Informally, to evaluate the time-complexity of the WEIGHTED COLORING Problem, the ideal way would be to reduce any 3-SAT formula  $\Phi$  to a weighted tree  $(T, w)$  such that (1) there is a correspondence between truth assignments of the variables of  $\Phi$  and the optimal colorings of  $T$ , and (2)  $\Phi$  is satisfiable if and only if  $\chi_w(T)$  is at most some pre-defined value  $M$  (depending on  $\Phi$ ). To do such a reduction, we would like to proceed as follows: given a boolean formula  $\Phi$  of size  $\eta$ , we build a tree  $T$  such that any truth assignment of  $\Phi$  for which  $\Phi$  is satisfied, we have a coloring of  $T$  of bounded weight, where the weight of a color reflects the truth assignment of a variable. However, proceeding that way, since the number of colors in an optimal weighted coloring of an  $n$ -node tree is at most  $O(\log n)$ ,  $T$  must have at least  $n = 2^\eta$  nodes. Hence, a polynomial-time algorithm to solve the WEIGHTED COLORING Problem in  $T$  would only lead to an exponential-time algorithm for deciding whether  $\Phi$  is satisfiable.

##### 3.1.2 From 3-SAT to INT-SAT.

To bypass this problem, we will use an auxiliary formula. Intuitively, given a 3-SAT formula with  $\eta$  boolean variables, we will translate it into another logical formula with  $\sqrt{\eta}$  *integral variables*. Using this new formula, we build a tree with  $2^{\sqrt{\eta}}$  nodes, where the weights of the colors in

coloring of bounded weight will correspond to the integral values of the variables. More formally,

**Definition 1** Given a set of  $n \times m$  boolean variables  $(y_j^i)_{i < n, j < m}$ , an integral assignment of these variables is a truth assignment such that, for any  $0 \leq i < n$ , at most one variable  $y_j^i$ ,  $\leq j < m$ , receives value 1.

A boolean formula  $\Phi$  with  $n \times m$  boolean variables  $(y_j^i)_{i < n, j < m}$  is integrally satisfiable w.r.t.  $(y_j^i)_{i < n, j < m}$  if there is an integral assignment of its variables that satisfies  $\Phi$ .

The INT-SAT Problem takes a formula  $\Phi$  with variables  $(y_j^i)_{i < n, j < m}$  as input and aims at deciding whether  $\Phi$  is integrally satisfiable w.r.t.  $(y_j^i)_{i < n, j < m}$ .

It is important to note that there is a one-to-one mapping between any integral assignment of a set of  $n \times m$  boolean variables  $(y_j^i)_{i < n, j < m}$  and the set of  $n$ -tuples  $(x_1, \dots, x_n)$  of integers in  $\{0, \dots, m\}$ . Indeed, for any  $i < n$ ,  $x_i = j$  if and only if  $y_j^i = 1$ , and  $x_i = 0$  if  $y_j^i = 0$  for any  $j < m$ .

We now show that 3-SAT can be sub-exponentially reduced to INT-SAT. This is an important ingredient of the proof of Proposition 2. We also think this result has its own interest and could be used in other context.

**Theorem 3** For any instance  $\Phi$  of 3-SAT with size  $\eta$ , there is a Boolean formula  $\Phi_{int}$  of size  $n = 2^{O(\sqrt{\eta})}$ , with variables  $(y_j^i)_{i < \sqrt{\eta}, j < 2^{\sqrt{\eta}}}$ , such that  $\Phi$  is satisfiable if and only if  $\Phi_{int}$  is integrally satisfiable w.r.t.  $(y_j^i)_{i, j}$ . Moreover,  $\Phi_{int}$  can be computed in time  $n$  and it is a CNF formula where all variables appear positively.

**Proof.** Let  $\Phi(u_1, \dots, u_\eta)$  be a instance of 3-SAT of size  $\eta = N^2$  (if  $\eta \neq N^2$ , we can add dummy variables). For any two integers  $a < N$  and  $b < 2^N$ , let  $bit(a, b)$  correspond to the  $a$ -th bit of the binary representation of  $b$ .

Let  $\Phi_{int}$  be the formula obtained from  $\Phi$  by replacing each literal  $u_{iN+j}$ ,  $0 \leq i < N$  and  $0 \leq j < N$ , by  $\bigvee_{\{\ell | bit(j, \ell)=1, 0 \leq \ell < 2^N\}} v_\ell^i$ . Then, each literal  $\bar{u}_{iN+j}$ ,  $0 \leq i < N$  and  $0 \leq j < N$  is replaced by  $\bigvee_{\{\ell | bit(j, \ell)=0, 0 \leq \ell < 2^N\}} v_\ell^i$ . Hence,  $\Phi_{int}$  has  $N \cdot 2^N$  variables  $(v_0^1, \dots, v_{2^N-1}^1, v_0^2, \dots, v_{2^N-1}^2, \dots, v_0^N, \dots, v_{2^N-1}^N)$  and  $poly(N)$  clauses of size  $O(2^N)$ . Because  $\Phi$  is in CNF, it is also the case for  $\Phi_{int}$ . Moreover, all variables appear positively in  $\Phi_{int}$ .

It remains to show that  $\Phi_{int}$  is integrally satisfiable if and only if  $\Phi$  is satisfiable.

First, let us assume that  $\Phi$  is satisfiable. Let  $u_1, \dots, u_\eta$  be a valid assignment of its variables and, for any  $0 \leq i < N$ , let  $x_i$  be the integer with  $(u_{N(i-1)+1}, \dots, u_{N(i-1)+N})$  as binary representation. Finally, for any  $i < N$  and  $j < 2^N$ , let us define  $v_j^i = 1$  if  $x_i = j$  and  $v_j^i = 0$  otherwise. By definition of  $\Phi_{int}$ ,  $(v_j^i)_{0 \leq i < N, 0 \leq j < 2^N}$  is a valid assignment and  $\Phi_{int}$  is therefore integrally satisfiable.

Conversely, let us assume that  $\Phi_{int}$  is integrally satisfiable and let  $(x_1, \dots, x_N)$  be  $N$  integers representing a valid assignment for it. Let  $u_1, \dots, u_\eta$  be defined such that, for any  $0 \leq i < N$ ,  $(u_{N(i-1)+1}, \dots, u_{N(i-1)+N})$  is the binary representation of  $x_i$ . Then,  $u_1, \dots, u_\eta$  is a valid assignment for  $\Phi$  which is satisfiable. ■

### 3.1.3 Proof of Proposition 2.

Hence, the above Theorem allows us to reduce any 3-SAT instance  $\Phi$  of size  $\eta$  into an INT-SAT instance  $\Phi_{int}$  with size  $2^{O(\sqrt{\eta})}$ . The key point is that this reduction allows us to turn the choice of  $\eta$  boolean variables into the choice of  $\sqrt{\eta}$  integers in  $\{0, \dots, 2^{\sqrt{\eta}}\}$ . Then, in further

sections, we build a tree  $T$  with  $2^{O(\sqrt{n})}$  vertices from the formula  $\Phi_{int}$ , such that there is a one to one mapping between any optimal weighted coloring of  $T$  and the  $\sqrt{n}$ -tuples of integers in  $\{0, \dots, 2^{\sqrt{n}}\}$ . Finally, our reduction ensures that the value of  $\chi_w(T)$  depends on the integral satisfiability of  $\Phi_{int}$  and therefore, on the satisfiability of  $\Phi$ . More formally, the next sections are devoted to prove the following result:

**Proposition 3** *For any CNF Boolean formula  $\Phi_{int}$  of size  $n$  where all variables  $(y_j^i)_{i,j}$  appear positively, there exist a weighted tree  $(T(\Phi_{int}), w)$  with a number of vertices polynomial in  $\mathcal{O}(n)$  and  $M \in \mathbb{R}$  such that  $\Phi_{int}$  is integrally satisfiable w.r.t.  $(y_j^i)_{i,j}$  if and only if  $\chi_w(T(\Phi_{int})) < M$ . Moreover,  $(T(\Phi_{int}), w)$  and  $M$  are computable in time polynomial in  $\mathcal{O}(n)$ .*

The proof of Proposition 2 is straightforward from Theorem 3 and Proposition 3.

### 3.2 Main ingredients of the proof of Proposition 3

In this section, we give the main ideas of the proof of Proposition 3. In particular, we describe the main property of the tree that we aim at building. The formal proof of it can be found in Section 4.

Let  $n, m \in \mathbb{N}$ . Let  $\Phi_{int}$  be any CNF Boolean formula with variables  $(y_j^i)_{i < n, j < m}$ . Let  $(T(\Phi_{int}), w)$  be a weighted tree, with  $\mathcal{O}(n \cdot m)$  nodes and weights can be encoded with  $\mathcal{O}(n \cdot m)$  bits. Let  $M \in \mathbb{R}$ , be a positive real value obtained from  $\Phi_{int}$ . The precise definitions of  $(T(\Phi_{int}), w)$  and  $M$  are given in next section.

Let  $\mathcal{C}$  be the set of all colorings of  $(T(\Phi_{int}), w)$  with weight at most  $M$ . We show in Section 4 that  $\chi_w(T(\Phi_{int})) \leq M$  and, therefore,  $\mathcal{C} \neq \emptyset$ .

#### 3.2.1 Weights structure and coloring constraints.

We ensure that the weights of the nodes of  $T(\Phi_{int})$  belong to some particular set that can be partitioned into  $\mathcal{O}(n)$  classes. Our construction ensures that, for any coloring in  $\mathcal{C}$  and for any class of weights, there is at most one color with weight in this class (see Corollary 1).

This property is very useful because it implies that, in any such coloring, each color is well identified by its weight. More precisely, any color of a coloring in  $\mathcal{C}$  is well identified as the unique one having weight in some interval. In particular, when building a coloring, we can affect a color  $c$  to a node  $v$  if the weight of  $v$  is not larger than the maximum weight that can take color  $c$  (and, of course, if  $v$  is not adjacent to any node with color  $c$ ).

The second important property ensured by the construction is that there is a correspondence between  $\mathcal{C}$ , i.e., the set of colorings of  $(T(\Phi_{int}), w)$  with weight at most  $M$ , and  $Y$ , the set of the all integral assignments of the boolean variables  $(y_j^i)_{i < n, j < m}$ . More precisely, we will show that:

- from any integral assignment  $y \in Y$ , if  $y$  is valid for  $\Phi_{int}$  (i.e.,  $\Phi_{int}(y)$  is true), then we can derive a coloring  $c_y$  of  $T(\Phi_{int})$  with weight strictly less than  $M$  (Lemma 6);
- on the other hand, for any coloring  $c$  with weight at most  $M$ , there is an integral assignment  $y_c \in Y$  such that  $\Phi(y_c)$  is true if and only if  $w(c) < M$  (Lemma 7).

This is the cornerstone of the proof of Proposition 3. Indeed, if  $\Phi_{int}$  is integrally satisfiable, then an integral assignment  $y \in Y$  that satisfies it (i.e., such that  $\Phi_{int}(y)$  is true) leads to a coloring  $c_y$  with weight strictly less than  $M$ . On the other hand, if  $\Phi_{int}$  is not integrally satisfiable, for any coloring  $c$  with weight at most  $M$ , then  $y_c$  does not satisfy  $\Phi_{int}$  and therefore  $w(c) \geq M$ . This means that, in that case, no coloring of weight strictly smaller than  $M$  exists.

### 3.3 A bit further into the proof of Proposition 3.

To conclude this section, we give some hints of the proof of the previous two properties.

#### 3.3.1 A tree per boolean variable.

The tree  $T(\Phi_{int})$  will be built from a family of subtrees  $(T(y_i^j))_{i < n, j < m}$ , each of which corresponds to a variable  $y_i^j$  ( $i < n$ ,  $j < m$ ) of the formula  $\Phi_{int}$ . Then, these subtrees will be combined in order to represent  $\Phi_{int}$ . More precisely, for any  $i < n$ ,  $j < m$ , and for each occurrence of  $y_i^j$  in  $\Phi_{int}$ , there will be a copy of a rooted tree  $T_i^j$  as a subtree of  $T(\Phi_{int})$ . We say that the root of such a subtree  $T_i^j$  is *representing* the variable  $y_i^j$ . Moreover, they will satisfy the following properties.

Let  $W_0$  and  $W_1$  be two particular colors that we will prove to be used in any coloring in  $\mathcal{C}$  (recall that, we can univocally identify the colors by their weight).

- First, let  $y \in Y$  be any integral assignment of the boolean variables  $(y_i^j)_{i < n, j < m}$  of  $\Phi_{int}$ . For any node  $v$  representing a variable  $y_i^j = 1$ , let  $c_v$  be arbitrarily chosen in  $\{W_0, W_1\}$  (possibly, two roots of two distinct (isomorphic) subtrees representing the same variable receive distinct colors). Then, there is a coloring  $c \in \mathcal{C}$  such that, for any  $v$  representing a variable  $y_i^j = 0$ ,  $c(v) = W_1$  and, for any  $v$  representing a variable  $y_i^j = 1$ ,  $c(v) = c_v$ . This means that we can color  $T(\Phi_{int})$ , without exceeding a weight of  $M$ , and such that, for any  $i < n$ ,  $j < m$  such that  $y_i^j = 1$ , we can arbitrary choose the color of each node representing  $y_i^j$  to be either  $W_0$  or  $W_1$ . Moreover, for any  $i < n$ ,  $j < m$  such that  $y_i^j = 0$ , the color of each node representing  $y_i^j$  is  $W_1$ .
- On the other hand, we will show that, for any coloring  $c \in \mathcal{C}$ , there exists an integral assignment  $y \in Y$  of the boolean variables  $(y_i^j)_{i < n, j < m}$  of  $\Phi$  such that, for any node  $v$  representing a variable  $y_i^j$ , then  $c(v) = W_1$  if  $y_i^j = 0$ , and  $c(v) \in \{W_0, W_1\}$  otherwise (Lemma 3). Intuitively, it means that any coloring of  $T(\Phi_{int})$  with weight at most  $M$  has a particular form that will be used to show that the coloring exceeds some weight when the formula is not satisfiable.

Intuitively, above properties say that a coloring of  $T(\Phi_{int})$  with weight at most  $M$  can be seen as an integral assignment of the variables. For any node  $v$  representing a variable, having a choice to color  $v$  means that the corresponding variable is assigned to true, while if the color of  $v$  is forced (to be  $W_1$ ), the corresponding variable is assigned to false.

#### 3.3.2 Combining the subtrees.

We now briefly describe how  $T(\Phi_{int})$  is obtained by combining the subtrees corresponding to the occurrences of the variables in  $\Phi_{int}$ . First, recall that  $\Phi_{int}$  is in Conjunctive Normal Form with all variables appearing positively. That is, each clause  $Q$  with  $|Q|$  variables is of the form  $\bigvee_{1 \leq k \leq |Q|} u_k$  where  $u_k \in \bigcup_{i < n, j < m} \{y_i^j\}$  for any  $k \leq |Q|$ . For any  $1 < k \leq |Q|$ , we recursively build a rooted subtree  $T(Q^k)$  from the rooted subtree  $T(Q^{k-1})$  and a copy of the rooted subtree representing  $u_k$ .

The key point is that, we can choose a coloring  $c \in \mathcal{C}$  with the color of the root of  $T(Q^k)$  being arbitrary  $W_0$  or  $W_1$  if and only if there is a coloring in  $\mathcal{C}$  where either the root of  $T(Q^{k-1})$  or the root of the subtree representing  $u_k$  can arbitrary have color  $W_0$  or  $W_1$ . Otherwise, the color of the root of  $T(Q^k)$  is forced to be  $W_1$ . That is, our construction simulates an OR where a true variable corresponds to a choice of a color, while a false variable corresponds to no choice.

By using this fact, the combination of these subtrees is done in such a way that we extend the properties of the subtrees representing the variables to the subtrees representing the clauses. Roughly, we ensure that, in any coloring  $c \in \mathcal{C}$  corresponding to an assignment  $y \in Y$ , the root of the subtree representing the clause  $Q$  must have color  $W_1$  if  $Q(y) = 0$  and can be arbitrary chosen in  $\{W_0, W_1\}$  otherwise.

To conclude, we have one tree per clause, such that, to any coloring in  $\mathcal{C}$  corresponds an integral assignment such that the color of the root of the clause-trees is forced to be  $W_1$  if the corresponding clause is false and can be arbitrary chosen in  $\{W_0, W_1\}$  otherwise. Finally, we connect all the roots of the clause-trees to a same node  $r$  whose color, in any coloring  $c \in \mathcal{C}$ , must be either  $W_1$  or a color  $W_3$  not used yet. Moreover, using  $W_3$  will imply that  $w(c) \geq M$ . Therefore, if at least one clause is false,  $W_3$  is forced in  $r$  and  $w(c) \geq M$ . Finally, if all clauses are true,  $r$  can be colored  $W_1$  and  $w(c) < M$ .

It is interesting to note that to optimally color the trees in the forest is easy and that the difficulty of the coloring of the final tree will arise from their combination.

## 4 Proof of Proposition 3

This section is devoted to the proof of Proposition 3.

**Proposition 3** *Let  $m = 2^n$ . For any CNF Boolean formula  $\Phi_{int}$  of size  $O(nm)$  where all variables  $(y_j^i)_{i \leq n, j \leq m}$  appear positively, there exist a weighted tree  $(T(\Phi_{int}), w)$  with size polynomial in  $O(nm)$  and  $M \in \mathbb{R}$  such that  $\Phi_{int}$  is integrally satisfiable w.r.t.  $(y_j^i)_{i,j}$  if and only if  $\chi_w(T(\Phi_{int})) < M$ .*

*Moreover,  $(T(\Phi_{int}), w)$  and  $M$  are computable in time polynomial in  $O(nm)$ .*

In order to prove Proposition 3, let us introduce some notations. Let  $n \in \mathbb{N}$  and let  $m = 2^n$ .

Let  $\Phi_{int}$  be a Boolean formula with  $n \times m$  variables  $\{y_i^j \mid 0 \leq i < n, 0 \leq j < m\}$  and  $L$  clauses, where  $L$  is polynomial in  $n$ . We assume that  $\Phi_{int}$  is in the Conjunctive Normal Form and that each variable appears positively. Moreover, we may assume that each variable appears at least once. That is,  $\Phi_{int} = \bigwedge_{\ell \leq L} Q_\ell$  and, for any  $\ell \leq L$ ,  $Q_\ell$  is the disjunction of  $p_\ell \geq 1$  positive variables.

Let  $\epsilon \in \mathbb{R}^*$  such that  $nm\epsilon = o(\frac{1}{2^{4n}})$ . Let

$$M = \sum_{i=0}^{4n+3} \frac{1}{2^i} + n(m-1)\epsilon < 2.$$

Let  $w_i^j = 1/2^i + j\epsilon$ , for any  $0 \leq i \leq 4n+3$  and  $0 \leq j \leq m$ . Let  $\mathcal{W} = \{w_i^j \mid 0 \leq i \leq 4n+3, 0 \leq j \leq m\}$  denote a set of weights. For any  $0 \leq k \leq 3$ , let  $W_k = w_{4i+k}^0 = 1/2^{4n+k}$ .

Finally, for any rooted tree  $T$ , let  $r(T)$  denote its root. A rooted tree  $S$  is a (*proper*) subtree of a rooted tree  $T$  if there is an edge  $e$  of  $T$  such that  $S$  is the connected component of  $T \setminus \{e\}$  that does not contain  $r(T)$ . We now define various subtrees required to build  $(T(\Phi_{int}), w)$ .

### 4.1 Binomial trees.

We first define a particular family of *binomial trees*  $B_i^j$ ,  $0 \leq i \leq 4n+2, 0 \leq j \leq m$ . They will be used in the construction of  $T(\Phi_{int})$ . Their role is to force the color of most of the nodes in any coloring  $c$  of  $T(\Phi_{int})$  with  $w(c) \leq M$ .

**Definition 2** *For any  $0 \leq i \leq 4n+2, 0 \leq j \leq m$ ,*

*let  $B_i^j$  be the weighted rooted tree defined recursively as follows (see Figure 2):*

- if  $i = 0$ , then  $B_0^j$  has a unique node with weight  $w_0^j$ ;
- otherwise,  $B_i^j$  has a root of weight  $w_i^j$  whose children are the roots of  $B_0^0, B_1^0, \dots, B_{i-1}^0$ .

Note that  $B_i^j$  has  $2^i$  nodes.

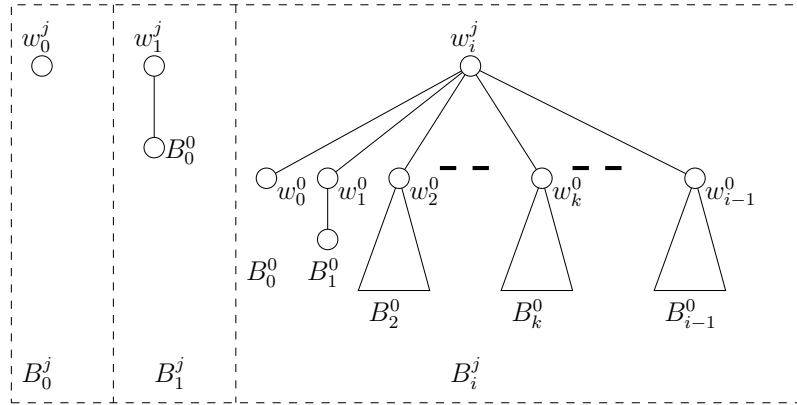


Figure 2: The construction of the binomial tree  $B_{i,j}$ .

**Lemma 1** Let  $0 \leq i \leq 4n+2, 0 \leq j \leq m$ . Let  $(T, w)$  be any weighted tree having  $B_i^j$  as subtree. If there exists a coloring  $c$  of  $(T, w)$  with  $w(c) \leq M$ , then, for any  $0 \leq k \leq i$ :

1. the vertices of  $B_i^j$  with weight in  $\{w_k^j \mid 0 \leq j \leq m\}$  receive the same color  $S_k$  of  $c$ ; and
2. there exists a unique color class  $S_k$  in  $c$  of weight in  $\{w_k^j \mid 0 \leq j \leq m\}$ .

**Proof.** The proof is by induction on the index  $i$ . In case  $i = 0$ , we prove both statements of the lemma at once by observing that any two vertices of  $(T, w)$  of weight in  $\{w_0^j \mid 0 \leq j \leq m\}$  must belong to the same color class  $S_0$ , otherwise we would conclude that  $w(c) \geq 2$ , that would be a contradiction to the fact that  $w(c) \leq M < 2$ .

Now, observe that the set of nodes of  $B_i^j$  with weight in  $\{w_k^j \mid 0 \leq j \leq m\}$  is an independent set that dominates the nodes with smaller weights (i.e., in  $\{w_{k'}^j \mid k < k' \leq i, 0 \leq j \leq m\}$ ).

By induction hypothesis, for any  $0 \leq k < i$ , the set of nodes of  $B_i^j$  with weight in  $\{w_k^j \mid 0 \leq j \leq m\}$  receive the same color  $S_k$  of  $c$  and this color class is the unique with weight in  $\{w_k^j \mid 0 \leq j \leq m\}$ . Then, for any  $0 \leq k < i$ , the root of  $B_i^j$  cannot be colored  $S_k$ , since it has a neighbor with weight  $w_k^0$ . Let  $S_i$  be color of the root of  $B_i^j$  in  $c$ . We prove that the color  $S_i$  cannot contain nodes with weight greater than  $w_{i-1}^{m-1}$  and that  $c$  cannot have another color  $S'_i \neq S_i$  with weight in  $\{w_i^j \mid 0 \leq j \leq m\}$ , because, otherwise the weight of  $c$  would be at least  $\frac{1}{2^i} + \sum_{k=0}^i \frac{1}{2^k} = 2 > M$  in both cases. ■

**Corollary 1** Let  $(T, w)$  be any weighted tree having  $B_{4n+2}^0$  as subtree. Let  $c$  be any coloring of  $(T, w)$  such that  $w(c) \leq M$ . Then,  $c = (S_0, \dots, S_k)$  with  $k \geq 4n+2$  and, for any  $0 \leq i \leq 4n+2$ ,  $S_i$  is the unique color with weight in  $\{w_i^j \mid 0 \leq j \leq m\}$ .

The trees we consider below will always satisfy the requirements of Corollary 1. Therefore, we are able to identify a color by its weight. In other words, in what follows, for any coloring  $c = (S_0, \dots, S_k)$  of weight at most  $M$  and for any  $i \leq 4n + 2$ ,  $S_i$  will be the unique color such that  $w(S_i) \in \{w_i^j \mid 0 \leq j \leq m\}$ .

Recall that we defined, for any  $0 \leq k \leq 3$ ,  $W_k = w_{4i+k}^0 = 1/2^{4n+k}$ . By a slight abuse of notation, for any  $0 \leq k \leq 3$ , we denote  $W_k = S_{4i+k}$  be the unique color with weight  $W_k$ .

## 4.2 Auxiliary trees and Variable-trees.

The family of *auxiliary trees*  $A_i^j$ ,  $0 \leq i < 4n, 0 \leq j \leq m$ , will be used to introduce some choice when coloring  $T(\Phi_{int})$ .

**Definition 3** For any  $0 \leq i < 4n, 0 \leq j \leq m$ ,

let  $A_i^j$  be the weighted rooted tree defined as follows (see Figure 3(a)):

1. let  $u$  be its root with weight  $w(u) = W_0$ , and connect it to a node  $v$  (its subroot) with weight  $w(v) = w_i^j$ ;
2.  $v$  is made adjacent to the root of a copy of  $B_\ell^0$ , for any  $0 \leq \ell < i - 1$ ;
3.  $u$  is made adjacent to the root of a copy of  $B_\ell^0$ , for any  $0 \leq \ell < 4n, \ell \neq i - 1$ .

Note that  $A_i^j$  consists of  $2^{4n}$  nodes.

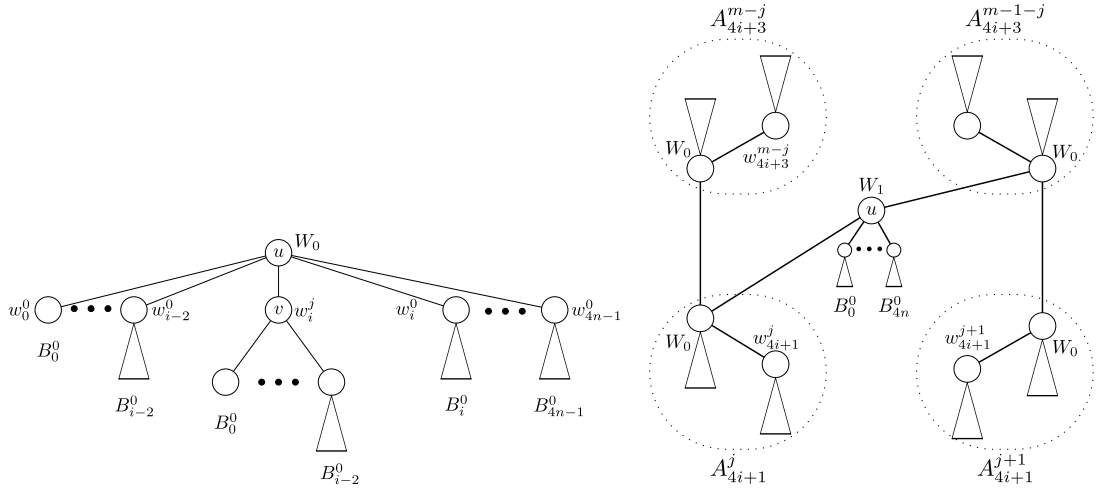


Figure 3: (a) (left) Auxiliary tree  $A_i^j$ . It is made from  $B_0^0, \dots, B_{i-2}^0, B_i^0, \dots, B_{4n-1}^0$  and a subtree of  $B_i^j$  (i.e.,  $B_i^j \setminus B_{i-1}^0$ ). (b) (right) The variable tree  $T(y_i^j)$ .

**Lemma 2** Let  $0 \leq i < 4n$  and  $0 \leq j \leq m$ . Let  $(T, w)$  be any weighted tree having  $B_{4n+2}^0$  and  $A_i^j$  as subtrees. Let  $u$  and  $v$  be the root and the sub-root of  $A_i^j$ , respectively. If there exists a coloring  $c$  of  $(T, w)$  with weight  $w(c) \leq M$ , then:

- either  $v$  is colored  $S_{i-1}$  and  $u$  must be colored with the color  $W_0$ ;
- or  $v$  is colored  $S_i$  (and therefore,  $w(S_i) \geq w_i^j$ ) and  $u$  can be colored with  $S_{i-1}$ .

**Proof.** Recall that, by Corollary 1, we can identify the colors of  $c$  and their weights. By Lemma 1, the root of each subtree  $B_k^0$ ,  $0 \leq k < 4n$ , must be colored with  $S_k$  and then the sub-root  $v$  can be colored only with color  $S_{i-1}$  or  $S_i$ . Note that, if  $v$  is colored with color  $S_p$  for some  $p > i$ , then  $w(S_p) \geq w_i^j$ , contradicting Corollary 1. In the first case,  $u$  is adjacent to a node with color  $S_k$ , for any  $k < 4n$ . Therefore,  $u$  must be colored with color  $S_{4n} = W_0$ . Otherwise,  $u$  may be colored with color  $S_{i-1}$ . ■

Intuitively, the previous lemma states that, either we “pay”  $j\epsilon$  in the weight of color  $S_i$ , or  $v$  must be colored with the color  $W_0$ . We now define the *variable-trees*  $T(y_i^j)$  using the auxiliary trees.

**Definition 4** For any  $0 \leq i < n, 0 \leq j < m$ ,

let  $T(y_i^j)$  be the weighted rooted tree, representing the variable  $y_i^j$ , defined as follows (see Figure 3(b)):

- let  $u$  be its root with weight  $w(u) = W_1$  and connected to the root of a copy of  $B_\ell^0$ , for any  $0 \leq \ell < 4n$ ;
- take one copy of  $A_{4i+1}^j, A_{4i+1}^{j+1}, A_{4i+3}^{m-j}$  and  $A_{4i+3}^{m-1-j}$  and:
  - connect  $r(A_{4i+1}^j)$  to  $r(A_{4i+3}^{m-j})$ , and  $r(A_{4i+1}^{j+1})$  to  $r(A_{4i+3}^{m-1-j})$ ;
  - connect  $u$  with  $r(A_{4i+1}^j)$  and  $r(A_{4i+3}^{m-j-1})$ .

Note that  $T(y_i^j)$  consists of  $O(2^{4n})$  nodes (i.e. polynomial in  $O(nm)$ ).

**Lemma 3** Let  $(T, w)$  be any weighted tree having  $B_{4n+2}^0$  as subtree and containing  $T(y_i^j)$  as subtree, for all  $0 \leq i < n$  and  $0 \leq j < m$ . Let  $c$  be a coloring of  $T$  with weight  $w(c) \leq M$ .

Then, there are  $(j_0, \dots, j_{n-1}) \in \{0, \dots, m-1\}^n$  such that each root  $u$  of each subtree  $T(y_i^{j_i})$ , for any  $0 \leq i < n$  and  $0 \leq j < m$ , satisfies:

- if  $j \neq j_i$ , then the color of  $u$  in  $c$  must be  $W_1$ ;
- otherwise, none of the two neighbors of  $u$  is colored  $W_0$  or  $W_1$ .

**Proof.** Since  $T$  contains  $B_{4n+2}^0$ , by Corollary 1, a coloring  $c = (S_0, \dots, S_k)$  of weight  $w(c) \leq M$  is such that  $k \geq 4n + 2$ , and, for any  $0 \leq i \leq 4n + 2$ ,  $S_i$  is the unique color such that  $w(S_i) \in \{w_k^j \mid 0 \leq j \leq m\}$ . In particular,  $w(c) \geq \sum_{i=0}^{4n+2} 1/2^i = M - n(m-1)\epsilon$ .

For any  $0 \leq i < n$ , let  $j_i \leq m$  be such that  $w(S_{4i+1}) = w_{4i+1}^{j_i}$ .

First, let us assume that  $j_i < m$ . In particular, this means that every sub-root of a subtree  $A_{4i+1}^r$ , for each  $j_i < r \leq m$ , is colored  $S_{4i}$  (recall that its color is either  $S_{4i}$  or  $S_{4i+1}$ , by Lemma 2). Consequently, any root of a subtree  $A_{4i+1}^r$ , for each  $j_i < r \leq m$ , must be colored  $W_0$ . Therefore, by the construction of the variable-trees, any root of a subtree  $A_{4i+3}^{m-r}$ , for each  $j_i < r \leq m$ , cannot be colored  $W_0$  because it is adjacent to a root of a subtree  $A_{4i+1}^r$ . Thus, by Lemma 2, it must be colored  $S_{4i+2}$  and the color of each sub-root of  $A_{4i+3}^{m-r}$  must be  $S_{4i+3}$ . Consequently,  $w(S_{4i+3}) \geq w_{4i+3}^{m-(j_i+1)}$ . Hence, for any  $0 \leq i < n$ , if  $j_i < m$ , we conclude that  $w(S_{4i+3}) + w(S_{4i+1}) \geq w_{4i+1}^{j_i} + w_{4i+3}^{m-(j_i+1)} = (m-1)\epsilon + 1/2^{4i+1} + 1/2^{4i+3}$ .

On the other hand, if  $j_i = m$ , it follows directly that  $w(S_{4i+3}) + w(S_{4i+1}) \geq m\epsilon + 1/2^{4i+1} + 1/2^{4i+3}$ .

Since  $w(c) \leq M$ , it implies that, for any  $0 \leq i < n$ ,  $j_i < m$  and  $w(S_{4i+3}) = w_{4i+3}^{m-j_i-1}$  and, for any  $0 \leq 2k < 4n$ ,  $w(S_{2k}) = w_{2k}^0$ . Consequently, by a similar argument, the roots of all subtrees



$A_{4i+3}^{m-j}$ , for each  $0 \leq j \leq j_i$ , must be colored  $W_0$  and, then, the roots of all subtrees  $A_{4i+1}^r$ , for each  $0 \leq j \leq j_i$ , must be colored  $S_{4i}$ .

Let  $0 \leq i < n$  and  $0 \leq j < m$ . Consider a subtree  $T(y_i^j)$  of  $T$ . If  $j \neq j_i$ , then (exactly) one of the roots of  $A_{4i+1}^j$  and  $A_{4i+3}^{m-1-j}$  must be colored  $W_0$ . In that case, the color of the root  $u$  of  $T(y_i^j)$  must be  $W_1$ . Indeed,  $u$  is adjacent to the root of  $B_k^0$ ,  $0 \leq k \leq 4n$ , and therefore it cannot be colored  $S_k$ . Moreover, if  $u$  is colored  $W_2$ , then we have a contradiction as  $w(c) > M$ , because  $w(u) = W_1$ . On the other hand, if  $j = j_i$ , none of the roots of  $A_{4i+1}^j$  and  $A_{4i+3}^{m-1-j}$  are colored  $W_0$  (for instance, if the root of  $A_{4i+1}^j$  is colored  $W_0$ , then the copy of  $A_{4i+3}^{m-1-j}$  that is adjacent to it would imply  $w(S_{4i+3}) \geq w_{4i+3}^{m-j_i-1}$ , a contradiction). Finally, none of the roots of  $A_{4i+1}^j$  and  $A_{4i+3}^{m-1-j}$  can be colored  $W_1$  because their weight is  $W_0$  (it would imply  $w(c) > M$ ). ■

### 4.3 Clause-trees and definition of $T(\Phi_{int})$ .

**Definition 5** Let  $Q_\ell = \bigvee_{1 \leq k \leq p_\ell} u_k$  be any clause of  $\Phi_{int}$  (recall that, for any  $1 \leq k \leq p_\ell$ ,  $u_k \in \{y_i^j \mid 0 \leq i < n, 0 \leq j < m\}$  and that  $\ell \leq L$ ).

For any  $1 \leq k \leq p_\ell$ , let  $T(Q_\ell^k)$  be the rooted weighted tree defined recursively as follows (see Figure 4(a)):

1.  $T(Q_\ell^1) = T(u_1)$ ;
2. for any  $k > 1$ , start with one copy of  $T(Q_\ell^{k-1})$  with root  $a$  and one copy of  $T(u_k)$  with root  $b$ .

Let  $c, d$  be two nodes with weight  $W_1$  and  $e, f$  be two nodes with weight  $W_2$ . For each node  $v \in \{a, b, c, d\}$ , and for any  $0 \leq i < 4n$ , add one copy of  $B_i^0$  and make its root adjacent to  $v$ . Add one copy of  $B_{4n+1}^0$  and make its root adjacent to  $e$ .

Finally, we add the edges  $\{\{a, f\}, \{b, c\}, \{c, f\}, \{d, e\}, \{e, f\}\}$  and  $d$  is chosen as the root.

Let us note  $T(Q_\ell) = T(Q_\ell^{p_\ell})$  the clause-tree corresponding to  $Q_\ell$  and that consists of  $O(p_\ell 2^{4n})$  nodes (i.e. polynomial in  $O(nm)$ ).

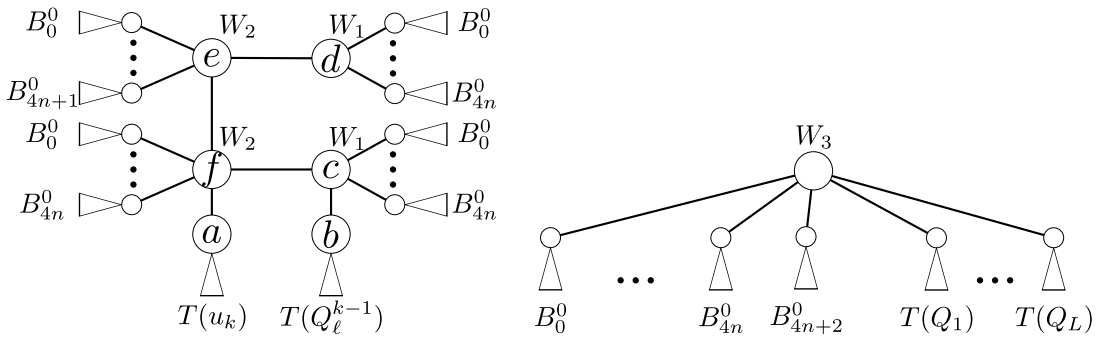


Figure 4: (a) (left) The clause tree  $T(Q_\ell^k)$ .

(b) (right) The final tree  $T(\Phi_{int})$ .

**Lemma 4** Let  $(T, w)$  be any weighted tree having  $B_{4n+2}^0$  as subtree and containing a subtree  $T(Q_\ell^k)$  as a subtree ( $\ell \leq L, k \leq p_\ell$ ). Let  $c$  be any coloring of  $T$  with weight  $w(c) \leq M$ . If  $a$  and  $b$  are colored  $W_1$ , then the color of the root  $d$  of  $T(Q_\ell^k)$  must be  $W_1$ ;

**Proof.** We prove it by induction on the number of variables  $k$  of  $Q_\ell^k$ . Observe that in case  $k = 1$ , then  $T(Q_\ell^k)$  is a variable-tree and the lemma trivially holds as the vertex  $b$  does not exist, thus the first statement is trivially satisfied, and, by Lemma 3, the color of its root must be either  $W_0$  or  $W_1$ .

Now, consider that  $a$  and  $b$  are roots of a variable-tree and of a clause-tree on  $k - 1$  variables  $T(Q_\ell^{k-1})$ , respectively. By Lemma 3 and by the inductive hypothesis, the colors of  $a$  and  $b$  are either  $W_0$  or  $W_1$ .

In case  $c(a) = c(b) = W_1$ , by the hypothesis  $w(c) \leq M$ , by Lemma 1 and Corollary 1, we conclude that  $c$  is colored  $W_0$ ,  $f$  is colored  $W_2$ ,  $e$  is colored  $W_0$  and  $d$  is forced to be colored  $W_1$ . This proves the first statement of the lemma. Finally, by the construction of  $T(Q_\ell^k)$ , by Lemma 1 and Corollary 1, the root  $d$  may be colored either  $W_0$  or  $W_1$ , since  $w(c) \leq M$ . ■

**Definition 6** Let  $T(\Phi_{int})$  be the weighted rooted tree obtained as follows (see Figure 4(b)). Let  $r$  be the root with weight  $W_3$ . For any  $1 \leq \ell \leq L$ , the root of one copy of  $T(Q_\ell)$  is made adjacent to  $r$ . For any  $0 \leq i \leq 4n + 2$ ,  $i \neq 4n + 1$ ,  $r$  is made adjacent to the root of one copy of  $B_i^0$ .

**Lemma 5**  $T(\Phi_{int})$  has size polynomial in  $m = 2^n$ .

**Proof.** Observe that each clause-tree  $T(Q_\ell)$  has size  $O(p_\ell 2^{4n})$  (see Definition 5), where  $p_\ell$  is polynomial in  $m$  (since  $p_\ell$  is at most the number  $nm$  of variables). Moreover, the number  $L$  of clauses is polynomial in  $m$  by the definition of  $\Phi_{int}$ . ■

**Lemma 6** If  $\Phi_{int}$  is integrally satisfiable, then  $\chi_w(T(\Phi_{int})) < M$ .

**Proof.** Let  $(y_i^j)_{i < n, j < m}$  be a valid integral assignment for  $\Phi_{int}$ . For any  $0 \leq i < n$ , let  $j_i$  be the (unique) index such that  $y_i^{j_i}$  is true. We construct a coloring  $c$  of  $(T(\Phi_{int}), w)$  such that  $w(c) < M$ .

By Lemma 1, in any coloring  $c$  of  $T(\Phi_{int})$  such that  $w(c) \leq M$ , the colors of all nodes of the binomial subtrees of  $T(\Phi_{int})$  are forced. Consequently, we only need to decide the colors of the following nodes: the roots and sub-roots of any copy of  $A_i^j$ , the roots of the trees  $T(y_i^j)$ , and the nodes added to connect the variables-trees into clause-trees (the nodes  $a, b, c, d, e, f$  in Figure 4(a)), for any  $0 \leq i < n$  and  $0 \leq j < m$ .

We first set the weight of color  $S_i$  for any  $0 \leq i < 4n$ . In particular, for any  $0 \leq i < n$ , the color  $S_{4i+1}$  must have weight  $w_{4i+1}^{j_i}$ . As we observed in the proof of Lemma 3, this choice fixes the colors of all roots and sub-roots of all the  $A_i^j$  trees, i.e. all the nodes in the variable trees, except to the roots of the variable-trees  $T(y_i^{j_i})$ , by Lemma 3.

More precisely, for any  $0 \leq i < n$  and  $0 \leq j < m$ , let us consider a subtree  $T(y_i^j)$ . Let  $j' \in \{j, j + 1\}$ . The sub-root of  $A_{4i+1}^{j'}$  receives color  $S_{4i+1}$  if  $j' \leq j_i$  and receives color  $S_{4i}$  otherwise. The root of  $A_{4i+1}^{j'}$  receives color  $S_{4i}$  if  $j' \leq j_i$  and receives color  $W_0$  otherwise. The sub-root of  $A_{4i+3}^{m-j'}$  receives color  $S_{4i+3}$  if  $j' > j_i$  and receives color  $S_{4i+2}$  otherwise. The root of  $A_{4i+3}^{m-j'}$  receives color  $S_{4i+2}$  if  $j' > j_i$  and receives color  $W_0$  otherwise. Finally, if  $j \neq j_i$ , the root of  $T(y_i^j)$  is colored  $W_1$ . On the other hand, if  $j = j_i$ , none of the neighbors of the root of  $T(y_i^j)$  is colored  $W_0$ , therefore, we can color it either  $W_0$  or  $W_1$ .

Now, let  $Q_\ell = \bigvee_{1 \leq k \leq p_\ell} u_k$  be any clause of  $\Phi_{int}$ . We will show that we can extend the previous coloring such that the root of the clause-tree  $T(Q_\ell)$  is colored  $W_0$  and the weight of the coloring is  $< M$ . This is by induction on  $p_\ell$ .

Indeed, if  $p_\ell = 1$ , then  $Q_\ell$  consists of a unique variable and this variable must be assigned to true (since the formula is true). Hence,  $Q_\ell = y_i^{j_i}$  for some  $0 \leq i < n$ . That is  $T(Q_\ell)$  is a subtree  $T(y_i^{j_i})$ . By paragraph above, we can color the root of it with  $W_0$ .

Now, assume that the result is correct for any clause of length  $p \geq 1$  and let  $p_\ell = p + 1$ . Thus,  $Q_\ell = u_{p+1} \vee (\vee_{1 \leq k \leq p} u_k)$ . Recall that  $T(Q_\ell)$  is built from a variable subtree  $T(u_{p+1})$  and a clause-subtree  $T(Q_\ell^p)$ . There are two cases to consider: either our assignment satisfies  $\vee_{1 \leq k \leq p} u_k$  or not. In the first case, the root of the clause-tree  $T(Q_\ell^p)$  (node  $b$  in Figure 4(a)) is colored  $W_0$  by induction. Moreover, by above paragraphs, the root of  $T(u_{p+1})$  (node  $a$  in Figure 4(a)) can be colored  $W_1$ . It is then easy to extend this coloring such that the root of  $T(Q_\ell)$  is colored  $W_0$ : in Figure 4(a), node  $c$  is colored  $W_1$ , node  $e$  is colored  $W_2$  and nodes  $f$  and  $d$  are colored  $W_0$ . If our assignment does not satisfy  $\vee_{1 \leq k \leq p} u_k$ , then it must satisfy  $u_{p+1}$ . That is,  $u_{p+1} = y_i^{j_i}$  for some  $0 \leq i < n$ . By a similar induction, we prove that the root of  $T(Q_\ell^p)$  can be colored  $W_1$ . Moreover, by above paragraphs, the root of  $T(u_{p+1}) = T(y_i^{j_i})$  can be colored  $W_0$ . It is then easy to extend this coloring such that the root of  $T(Q_\ell)$  is colored  $W_0$ : in Figure 4(a), node  $f$  is colored  $W_1$ , node  $e$  is colored  $W_2$  and nodes  $c$  and  $d$  are colored  $W_0$ .

Thus, we color the roots of all the clause-trees with color  $W_0$  and the root of  $T(\Phi_{int})$  with the color  $W_1$ .

Consequently, the weight of this coloring  $c$  is  $w(c) = \sum_{i=0}^{4n+2} \frac{1}{2^i} + n(m-1)\epsilon < M$ . ■

**Lemma 7** *If  $\Phi_{int}$  is integrally satisfiable, then  $\chi_w(T(\Phi_{int})) \geq M$ .*

**Proof.**  $\Phi_{int}$  is not integrally satisfiable Let  $c$  be a coloring of  $T(\Phi_{int})$  with weight at most  $M$ .

By Lemma 3, there are integers  $(j_0, \dots, j_{n-1})$  such that the color of the root of any subtree  $T(y_i^{j_i})$  is forced to be  $W_1$ , if  $j \neq j_i$ . Let  $Y = (y_i^j)_{i < n, j < m}$  be the corresponding integral assignment. In other words, for any variable  $y_i^j$  ( $0 \leq i < n, 0 \leq j < m$ ),  $y_i^j = 0$  if  $j \neq j_i$ .

Since  $\Phi_{int}$  is not integrally satisfiable, there is a clause  $Q$  that is not satisfied by this assignment. Let us consider the clause-subtree  $T(Q)$ . It has been built from variable-trees corresponding to the variables constituting the clause  $Q$ . Because all these variables are assigned to false, the roots of these variable-trees are all colored with  $W_1$ , by Lemma 3.

By induction on the length of  $Q$  and by Lemma 4, the color of the root of  $T(Q_\ell)$  must be  $W_1$ . Thus, the root of  $T(\Phi_{int})$  can just be colored  $W_3$ . Consequently, the coloring  $c$  has weight  $w(c) \geq \sum_{i=0}^{4n+3} \frac{1}{2^i} + n(m-1)\epsilon = M$ . ■

Proposition 3 follows directly from Lemmas 5, 6 and 7.

## References

- [1] Araujo, J., Nisse, N., Pérennes, S.: Weighted coloring in trees, [http://www-sop.inria.fr/members/Julio-Cesar.Silva\\_Araujo/weightedtree.pdf](http://www-sop.inria.fr/members/Julio-Cesar.Silva_Araujo/weightedtree.pdf)
- [2] Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC). pp. 151–158 (1971)
- [3] Demange, M., de Werra, D., Monnot, J., Paschos, V.T.: Weighted node coloring: When stable sets are expensive. In: 28th International Workshop on Graph-Theoretic Concepts in Computer Science (WG). LNCS, vol. 2573, pp. 114–125. Springer (2002)

- 
- [4] Escoffier, B., Monnot, J., Paschos, V.T.: Weighted coloring: further complexity and approximability results. *Inf. Process. Lett.* 97(3), 98–103 (2006)
  - [5] Guan, D.J., Zhu, X.: A coloring problem for weighted graphs. *Inf. Process. Lett.* 61(2), 77–81 (1997)
  - [6] Impagliazzo, R., Paturi, R.: On the complexity of  $k$ -sat. *J. Comput. Syst. Sci.* 62(2), 367–375 (2001)
  - [7] Karp, R.M.: Reducibility among combinatorial problems. In: *Proceedings of a symposium on the Complexity of Computer Computations*. pp. 85–103. The IBM Research Symposia Series, Plenum Press, New York (1972)
  - [8] Kavitha, T., Mestre, J.: Max-coloring paths: tight bounds and extensions. *J. Comb. Optim.* 24(1), 1–14 (2012)
  - [9] Linhares Sales, C., Reed, B.: Weighted coloring on graphs with bounded tree width. In: *Annals of 19th International Symposium on Mathematical Programming* (2006)
  - [10] de Werra, D., Demange, M., Escoffier, B., Monnot, J., Paschos, V.T.: Weighted coloring on planar, bipartite and split graphs: Complexity and approximation. *Discrete Applied Mathematics* 157(4), 819–832 (2009)



**RESEARCH CENTRE  
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93  
06902 Sophia Antipolis Cedex

Publisher  
Inria  
Domaine de Voluceau - Rocquencourt  
BP 105 - 78153 Le Chesnay Cedex  
[inria.fr](http://inria.fr)

ISSN 0249-6399