HAL
open science

b-coloring is NP-hard on co-bipartite graphs and polytime solvable on tree-cographs

Flavia Bonomo, Oliver Schaudt, Maya Stein, Mario Valencia-Pabon

To cite this version:

Flavia Bonomo, Oliver Schaudt, Maya Stein, Mario Valencia-Pabon. b-coloring is NP-hard on co-bipartite graphs and polytime solvable on tree-cographs. Algorithmica, 2014, pp.17. 10.1007/s00453-014-9921-5 . hal-01102516

HAL Id: hal-01102516

https://hal.science/hal-01102516v1

Submitted on 12 Jan 2015

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.

b-coloring is NP-hard on co-bipartite graphs and polytime solvable on tree-cographs *

Flavia Bonomo[1], Oliver Schaudt[2], Maya Stein[3], and Mario Valencia-Pabon[†4]

[1]CONICET and Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina. Email: fbonomo@dc.uba.ar
[2]Institut de Mathématiques de Jussieu, CNRS UMR7586, Université Pierre et Marie Curie (Paris 6), Paris, France. Email: schaudt@math.jussieu.fr
[3]Centro de Modelamiento Matemático, Universidad de Chile, Santiago, Chile. Email: mstein@dim.uchile.cl
[4]Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR7030, Villetaneuse, France. Email: valencia@lipn.univ-paris13.fr

Abstract

A *b-coloring* of a graph is a proper coloring such that every color class contains a vertex that is adjacent to all other color classes. The *b-chromatic number* of a graph G, denoted by $\chi_b(G)$, is the maximum number t such that G admits a b-coloring with t colors. A graph G is called *b-continuous* if it admits a b-coloring with t colors, for every $t = \chi(G), \ldots, \chi_b(G)$, and *b-monotonic* if $\chi_b(H_1) \geq \chi_b(H_2)$ for every induced subgraph H_1 of G, and every induced subgraph H_2 of H_1.

We investigate the b-chromatic number of graphs with stability number two. These are exactly the complements of triangle-free graphs, thus including all complements of bipartite graphs. The main results of this work are the following:

1. We characterize the b-colorings of a graph with stability number two in terms of matchings with no augmenting paths of length one or three. We derive that graphs with stability number two are b-continuous and b-monotonic.
2. We prove that it is NP-complete to decide whether the b-chromatic number of co-bipartite graph is at least a given threshold.
3. We describe a polynomial time dynamic programming algorithm to compute the b-chromatic number of co-trees.

*Partially supported by UBACyT Grant 20020100100980, CONICET PIP 112-200901-00178 and 11220120100450CO, and ANPCyT PICT 2012-1324 (Argentina) and MathAmSud Project 13MATH-07 (Argentina–Brazil–Chile–France).

†Currently in "Délégation" at INRIA Nancy - Grand Est, 2013-2015.

4. Extending several previous results, we show that there is a polynomial time dynamic programming algorithm for computing the b-chromatic number of tree-cographs. Moreover, we show that tree-cographs are b-continuous and b-monotonic.

1 Introduction

A *b-coloring* of a graph G by k colors is a proper coloring of the vertices of G such that every color class contains a vertex that is adjacent to all the other $k-1$ color classes. Such a vertex will be called a *dominating vertex*. It is easy to see that any proper coloring of a graph G with $\chi(G)$ many colors is a b-coloring (as usual, we denote by $\chi(G)$ the minimum number of colors needed for a proper coloring of the vertices of a graph).

The *b-chromatic number* of a graph G, denoted by $\chi_b(G)$, is the maximum number k such that G admits a b-coloring with k colors. Clearly, $\chi_b(G) \leq \Delta(G) + 1$ where $\Delta(G)$ denotes the maximum degree of G. The b-chromatic number was introduced in [11]. The motivation, similarly as the well known achromatic number (cf. e.g, [7, 2] and ref. therein), comes from algorithmic graph theory. Suppose one colors a given graph properly, but in an arbitrary way. After all vertices are colored, one would wish to perform some simple operations to reduce the number of colors. A simple operation consists in recoloring all the vertices in one color class with a possible different color. Obviously, such recoloring is impossible if each color class contains a dominating vertex. Hence, the b-chromatic number of the graph serves as the tight upper bound for the number of colors used by this coloring heuristic. From this point of view, both complexity results and polynomial time algorithms for particular graph families are interesting.

Assume that the vertices $v_1, v_2, \ldots, v_n$ of a graph G are ordered such that $d(v_1) \geq d(v_2) \geq \ldots \geq d(v_n)$, where $d(x)$ denotes the degree of vertex x in G. Let

$$m(G) := \max\{i : d(v_i) \geq i - 1\}$$

be the maximum number i such that G contains at least i vertices of degree $\geq i - 1$. It is clear that $m(G) \leq \Delta(G) + 1$. Irving and Manlove [11] showed that this parameter bounds the b-chromatic number:

Proposition 1. *For every graph G, $\chi(G) \leq \chi_b(G) \leq m(G)$.*

Irving and Manlove [11] also showed that determining $\chi_b(G)$ is NP-complete for general graphs, but polynomial-time solvable for trees. Kratochvíl, Tuza and Voigt [13] proved that the problem of determining if $\chi_b(G) = m(G)$ is NP-complete even for connected bipartite graphs G with $m(G) = \Delta(G) + 1$. A graph G is *tight* if it has exactly $m(G)$ *dense* vertices (a vertex v of a graph G is dense if $d(v) \geq m(G) - 1$), each of which has degree exactly $m(G) - 1$. Havet, Linhares-Sales and Sampaio [8] recently

investigated the problem on tight graphs. They proved that the problem of determining if a tight graph G has $\chi_b(G) = m(G)$ is NP-complete for bipartite graphs and ptolemaic graphs, but polynomial-time solvable for complements of bipartite graphs, split graphs and block graphs.

In last years, several related concepts concerning b-colorings of graphs have been studied in [6, 8, 9, 10, 12]. A graph G is defined to be *b-continuous* [6] if it admits a b-coloring with t colors, for every $t = \chi(G), \ldots, \chi_b(G)$. In [12] (see also [6]) it is proved that chordal graphs and some planar graphs are b-continuous. A graph G is defined to be *b-monotonic* [3] if $\chi_b(H_1) \geq \chi_b(H_2)$ for every induced subgraph H_1 of G, and every induced subgraph H_2 of H_1. They proved that P_4-sparse graphs (and, in particular, cographs) are b-continuous and b-monotonic. Besides, they give a dynamic programming algorithm to compute the b-chromatic number in polynomial time within these graph classes.

Our paper is organized as follows. In the next section, we characterize b-colorings of graphs with stability number two in terms of matchings with no augmenting paths of length one or three.

In Section 3, we prove that graphs with stability at most two are both b-continuous and b-monotonic.

In Section 4, we prove that computing the b-chromatic number of co-bipartite graphs is an NP-complete problem.

Finally, in Section 5, first we describe a polynomial-time dynamic programming algorithm to compute the b-chromatic number of co-trees. Next, we extend our results to the family of tree-cographs by showing that there is a polynomial time dynamic programming algorithm for computing the b-chromatic number of graphs in this family and that these are also b-continuous and b-monotonic.

2 b-colorings and matchings

The *stability* of a graph G is defined as the maximum cardinality of a subset of pairwise non-adjacent vertices in G. Given a graph G, we denote by $\overline{G}$ the complement graph of G, which is the graph on the same set of vertices as G that has an edge between two different vertices u and v if and only if u and v are non-adjacent in G. It is not difficult to see that G is a graph with stability one if and only if it is complete, and G is a graph with stability at most two if and only if $\overline{G}$ is a triangle-free graph. In this section, we will see that matchings in triangle-free graphs are very important when we deal with b-colorings of graphs with stability at most two.

Let M be a matching of a graph G. Denote by $V(M)$ the set of all vertices covered by M. An *augmenting path* for M is a path starting and ending outside $V(M)$ whose edges alternate between $E(G) - M$ and M. Usually,

M is called *maximal* if no further edge can be included in M. In other words, G does not contain an augmenting path of length one with respect to M. Following this terminology we call M *strongly maximal* if G does not contain augmenting paths of length one or three with respect to M. Trivially, maximum matchings are strongly maximal, and strongly maximal matchings are maximal. Our next lemma shows why strongly maximal matchings are important in our setting.

Lemma 2. *Let G be a graph of stability at most two and let c be a proper coloring of G. Then c is a b-coloring if and only if the set*

$$M = \{uv : u, v \in V, u \neq v \text{ and } c(u) = c(v)\}$$

is a strongly maximal matching in $\overline{G}$. Moreover, the number of colors c uses is $|V(G)| - |M|$.

Proof. First, observe that M is a (possibly empty) matching of $\overline{G}$ because G has stability at most two. Now, suppose that $\overline{G}$ contains an augmenting path P of length 1 or 3 for M. If P consists of only one edge uv, then in G, the vertices u and v are non-adjacent, and each makes up a singleton color class. Thus c is not a b-coloring. If P has three edges, say $x - u - v - y$, then for each of the endvertices of its middle edge uv there is a singleton color class with no vertex adjacent to it in G. In fact, $\{u, v\}$, $\{x\}$ and $\{y\}$ correspond to three different color classes in the coloring c, but there is no vertex in $\{u, v\}$ adjacent to both vertices x and y. So the color class $\{u, v\}$ witnesses the fact that c is not a b-coloring.

Next, suppose that c is not a b-coloring. Note that, as G has stability at most two, every vertex of G is adjacent (in G) to at least one vertex of any given color class of size 2. So, the witness for c not being a b-coloring is one of the following two: either it is a singleton color class whose vertex is non-adjacent to another singleton color class, or it is a color class $\{u, v\}$ of size two, such that u is non-adjacent to some singleton color class, and v is non-adjacent to a different singleton color class. Clearly, the first situation corresponds to an augmenting path of M on one edge, and the second situation corresponds to an augmenting path of M on three edges. □

Observe that coloring c from Lemma 2 is a maximum (minimum) b-coloring of G if and only if M is a minimum (maximum) strongly maximal matching of $\overline{G}$.

3 b-continuity and b-monotonicity of graphs with stability at most two

In order to prove the b-continuity of graphs with stability at most two, we need the following result.

Lemma 3. *Let M be a strongly maximal matching of a graph G and let P be a minimum length augmenting path in G with respect to M. Then, the matching $M' = (M \setminus E(P)) \cup (E(P) \setminus M)$ is a strongly maximal matching of G, and $|M'| = |M| + 1$.*

Proof. Let $P = (x_1, x_2, \ldots, x_k)$. By basic results from matching theory, the only thing we need to prove is that M' is again strongly maximal. Since the maximality of M' is clear, suppose for contradiction that there is an augmenting path of length 3, say $Q = (u, v, w, x)$. Necessarily vw is an edge of $M' \setminus M$, and thus w.l.o.g. there is some $i \in \{1, 2, \ldots, k-1\}$ with $v = x_i$ and $w = x_{i+1}$. Moreover, $u, x \notin V(M)$. Thus both paths $(x_1, x_2, \ldots, x_i, u)$ and $(x, x_{i+1}, x_{i+2}, \ldots, x_k)$ are augmenting paths for M and at least one of these paths is shorter than P. This is a contradiction to the choice of P. $\square$

By Lemma 2, any b-coloring using $k > \chi(G)$ colors of a graph G of stability at most two corresponds to a strongly maximal matching M that is not maximum. By Berge's lemma [1], there is an augmenting path for M. Using Lemma 3 we obtain a strongly maximal matching M' of cardinality $|M| + 1$, which, again by Lemma 2, corresponds to a b-coloring with $k - 1$ colors. Repeatedly applying this argument gives the following result.

Theorem 4. *Graphs of stability at most two are b-continuous.*

Given a maximum b-coloring of a graph G of stability at most two, we can thus find b-colorings for all values between $\chi(G)$ and $\chi_b(G)$. Moreover, we can do this in polynomial time, provided we can find a minimum length augmenting path for a given matching in polynomial time. This is the aim of the following lemma that can be derived by a slight modification of Edmonds' blossom algorithm [5].

Lemma 5. *Let M be a matching in a graph G. Then, a minimum length augmenting path P in G with respect to M can be computed in polynomial time.*

Lemma 5 together with the proof of Theorem 4 implies that given a graph G of stability at most two, and a b-coloring of G using $k > \chi(G)$ colors, we can compute in polynomial time a b-coloring for G with $k - 1$ colors. Notice that the converse is not necessarily true, i.e., if we have a b-coloring of G using $k < \chi_b(G)$ colors, we do not know how to compute in polynomial time a b-coloring for G with $k + 1$ colors. Indeed, we will prove in the next section that the problem of computing the b-chromatic number of a graph with stability at most two is NP-complete, even when restricted to the smaller class of co-bipartite graphs.

We now turn to the b-monotonicity of graphs of stability at most two.

Theorem 6. *Graphs of stability at most two are b-monotonic.*

Proof. The class of graphs of stability at most two is closed under taking induced subgraphs. Thus we only have to prove that χ_b is monotonously decreasing under the deletion of a vertex. In view of Lemma 2, it is sufficient to show that given a graph G of stability 2 and some vertex $v \in V(G)$ the following holds: If there is a strongly maximal matching of $\overline{G-v}$ of size k, then there is a strongly maximal matching of $\overline{G}$ of size at most $k+1$. This implies $\chi_b(G) \geq \chi_b(G-v)$.

Let M be a strongly maximal matching of $\overline{G-v}$. We now consider M as a matching of $\overline{G}$. If M is a strongly maximal matching of $\overline{G}$, we are done. So we assume that M is not strongly maximal, and thus there is an augmenting path P of length at most 3. We may choose P of minimum length among all augmenting paths of M in $\overline{G}$. Note that P meets v, say P starts in v.

We assume first that P is of length 1, that is, v has an unmatched neighbor in $\overline{G}$, say u. We claim the matching $M' = M \cup \{uv\}$ is strongly maximal, in which case we are done. Indeed, otherwise there is an augmenting path of length 3 for M', and uv is the central edge of this path. So, there is a neighbor of u, say w, that is not matched by M'. In particular, $v \neq w$. Thus uw is disjoint from M, contradicting the maximality of M in $\overline{G-v}$. This proves our claim.

Now assume that P is of length 3, say $P = (v, u, w, x)$. Let $M' = (M \setminus \{uw\}) \cup \{vu, wx\}$. Suppose that M' is not strongly maximal in $\overline{G}$. Then there is an augmenting path of length 3, with central edge either vu or wx. In either case, x or v has a neighbor in $\overline{G}$ that is unmatched by M' and thus also by M, a contradiction either to the maximality of M in $\overline{G-v}$ or to the minimality of P. This completes the proof. □

4 NP-hardness result for co-bipartite graphs

As mentioned in Section 1, Havet, Linhares-Sales and Sampaio [8] proved that the problem of determining if a tight co-bipartite graph G has $\chi_b(G) = m(G)$ is polynomial-time solvable. However, the computational complexity of χ_b in the class of co-bipartite graphs is left open. In the next theorem, we prove that b-coloring general co-bipartite graphs is a hard problem.

Theorem 7. *Given a co-bipartite graph G and a natural number k, it is NP-complete to decide whether G admits a b-coloring with at least k colors.*

Proof. By Lemma 2, it suffices to prove that it is NP-complete to decide whether a bipartite graph G admits a strongly maximal matching containing at most k edges, when G and k are given input.

Our reduction is from the minimum maximal matching problem which is to decide whether a given graph admits a maximal matching of at most k edges, for given k. This problem is NP-complete even if the instances are restricted to bipartite graphs, as shown by Yannakakis and Gavril [15].

Given a bipartite graph G with m edges, we define a new graph H_G as follows. For each edge $uv \in E(G)$ we introduce a set of new vertices

$$X_{uv} = \{x^1_{uv}, x^2_{uv}, x^3_{uv}, x^4_{uv}, x^1_{vu}, x^2_{vu}, x^3_{vu}, x^4_{vu}\}$$

and edges

$$F_{uv} = \{ux^1_{uv}, x^1_{uv}x^2_{uv}, x^2_{uv}x^3_{uv}, x^3_{uv}x^4_{uv}, x^1_{uv}x^1_{vu}, x^1_{vu}x^2_{vu}, x^2_{vu}x^3_{vu}, x^3_{vu}x^4_{vu}, vx^1_{vu}\}.$$

Note that $X_{uv} = X_{vu}$ and $F_{uv} = F_{vu}$. Then H_G is defined by

$$V(H_G) = V(G) \cup \bigcup_{uv \in E(G)} X_{uv},$$

$$E(H_G) = \bigcup_{uv \in E(G)} F_{uv}.$$

Clearly H_G can be computed in polynomial time. Moreover, H_G is bipartite since G is. For each edge $uv \in E(G)$, we define the following auxiliary sets of edges in H_G:

$$F^{\in}_{uv} = \{ux^1_{uv}, x^2_{uv}x^3_{uv}, x^2_{vu}x^3_{vu}, vx^1_{vu}\} \text{ and } F^{\notin}_{uv} = \{x^1_{uv}x^1_{vu}, x^2_{uv}x^3_{uv}, x^2_{vu}x^3_{vu}\}$$

We claim the following:

Claim 8. *There exists a minimum strongly maximal matching M of H_G such that*

$$x^3_{uv}x^4_{uv} \notin M \text{ for each edge } uv \in E(G).$$

Moreover, M can be obtained from any minimum strongly maximal matching of H_G in polynomial time.

In order to prove this claim, we proceed by contradiction. Assume that every minimum strongly maximal matching of H_G contains at least an edge $x^3_{uv}x^4_{uv}$ for some edge $uv \in E(G)$, and let M be a minimum strongly maximal matching of H_G having a minimum number of edges of the form $x^3_{uv}x^4_{uv}$. Note that the choice of M implies that for every edge $uv \in E(G)$ we have that

(*i*) $x^3_{uv}x^4_{uv} \in M$ if and only if $x^1_{uv}x^2_{uv} \in M$. If $x^1_{uv}x^2_{uv} \in M$ then $x^3_{uv}x^4_{uv} \in M$, otherwise, M is not maximal. If $x^3_{uv}x^4_{uv} \in M$ then $x^1_{uv}x^2_{uv} \in M$, otherwise, we could replace $x^3_{uv}x^4_{uv}$ by $x^2_{uv}x^3_{uv}$ in M (the resulting matching is strongly maximal as M is so), contradicting the choice of M.

(*ii*) If the edges $x^3_{uv}x^4_{uv}$ and $x^1_{uv}x^2_{uv}$ are in M, then we have that vertices u and x^1_{vu} are each matched by M. Otherwise, if u is unmatched, we can replace $x^1_{uv}x^2_{uv}, x^3_{uv}x^4_{uv} \in M$ with the edges $x^2_{uv}x^3_{uv}, ux^1_{uv}$. This again yields a strongly maximal matching (since u has no neighbors unmatched by M), contradicting the choice of M. We can use the same argument in the case x^1_{vu} is unmatched.

This is also some of the steps in order to transform any minimum strongly maximal matching into the desired one.

Now, let uv be and edge in the graph G such that $x^3_{uv}x^4_{uv} \in M$. By (i) and (ii), we can deduce that $|M \cap F_{uv}| = 4$. Consider the matching

$$\tilde{M} := (M \setminus F_{uv}) \cup F^{\notin}_{uv}$$

We claim that $\tilde{M}$ is strongly maximal. As $\tilde{M}$ is smaller than M, we thus obtain the desired contradiction.

So assume $\tilde{M}$ is not strongly maximal. Then, as u is matched, there is an augmenting path P of length 1 or 3 starting at v.

Now, observe that all neighbors of v are of the form x^1_{vw} (for some $w \in V(G)$), and thus, as neither $x^1_{vw}x^2_{vw}$ nor $x^1_{vw}x^2_{vw}x^3_{vw}x^4_{vw}$ is an augmenting path for the strongly maximal matching M, all neighbors of v are matched by M.

So, P has length 3, and it is easy to see that P has to end in some (unmatched) vertex $w \in V(G) \setminus \{u, v\}$ (by the maximality of M, every vertex x^3_{wz} is matched by M, and by the choice of M, every vertex x^2_{wz} is matched by M). By (i) and (ii), we know that $F_{vw} \cap M = F^{\notin}_{vw}$. Consider the matching

$$(\tilde{M} \setminus F^{\notin}_{vw}) \cup F^{\in}_{vw}.$$

This matching is clearly strongly maximal, and has fewer edges of the form $x^3_{uv}x^4_{uv}$, contradicting the choice of M. (And this is the remaining step in order to transform any minimum strongly maximal matching into the desired one.) This ends the proof of Claim 8.

Therefore, by Claim 8, we have that there is a minimum strongly maximal M' in H_G that verifies either $F_{uv} \cap M' = F^{\in}_{uv}$ or $F_{uv} \cap M' = F^{\notin}_{uv}$ for each edge $uv \in E(G)$.

Next we show that if M is a minimum maximal matching of G and M' is a minimum strongly maximal matching of H_G, $|M| = |M'| - 3m$. As explained above, this completes the proof.

Let M be a minimum maximal matching of G. Using the auxiliary sets $F^{\in}_{uv}$ and $F^{\notin}_{uv}$, we define a strongly maximal matching M' of H_G by

$$M' = \bigcup_{uv \in M} F^{\in}_{uv} \cup \bigcup_{uv \notin M} F^{\notin}_{uv}.$$

Note that $|M'| = |M| + 3m$.

Now, let M' be a minimum strongly maximal matching of H_G that verifies either $F_{uv} \cap M' = F^{\in}_{uv}$ or $F_{uv} \cap M' = F^{\notin}_{uv}$ for each edge $uv \in E(G)$. We define a maximal matching M of G by setting

$$M = \{uv : uv \in E(G),\ F_{uv} \cap M' = F^{\in}_{uv}\}.$$

Clearly $|M| = |M'| - 3m$, which completes the proof. $\square$

5 b-coloring co-trees and tree-cographs

5.1 co-trees

Theorem 9. *In the class of co-trees, χ_b can be computed in polynomial time.*

Proof. According to Lemma 2, the problem is equivalent to find a minimum strongly maximal matching (MSMM) in a tree. We will do it for nontrivial trees by dynamic programming. In order to do so, we will define five functions $F_i(r, s)$, $i = 1, \ldots, 5$, for a nontrivial tree T_{rs} rooted at a leaf r with neighbor s. As we will apply them to the subtrees of a tree, we will assume that r can have neighbors outside T_{rs}.

- $F_1(r, s)$: cardinality of a MSMM of T_{rs} such that r is unmatched, and ∞ if it does not exist.
- $F_2(r, s)$: cardinality of a MSMM of T_{rs} that uses the edge rs and such that s may or may not have an unmatched neighbor (this case will apply when r has no unmatched neighbor outside T_{rs}), and ∞ if it does not exist.
- $F_3(r, s)$: cardinality of a MSMM of T_{rs} that uses the edge rs and such that s cannot have an unmatched neighbor (this case will apply when r has already an unmatched neighbor outside T_{rs}, so an unmatched neighbor of s will complete an augmenting path of length 3 in the whole tree), and ∞ if it does not exist.
- $F_4(r, s)$: cardinality of a MSMM of T_{rs} such that the vertex s is matched with some vertex different from r and the vertex r is considered as "already matched" (this case will apply when r is already matched with a vertex outside T_{rs}), and ∞ if it does not exist.
- $F_5(r, s)$: cardinality of a MSMM of T_{rs} such that the vertex s remains unmatched and the vertex r is considered as "already matched", and ∞ if it does not exist.

With these definitions, for the base case in which $V(T_{rs}) = \{r, s\}$, we have

- $F_1(r, s) = \infty$ (if r is unmatched and s has no further neighbors, the matching will never be maximal)
- $F_2(r, s) = 1$ (precisely, the edge rs)
- $F_3(r, s) = 1$ (precisely, the edge rs)
- $F_4(r, s) = \infty$ (it is not feasible because s has no further neighbors)

- $F_5(r,s) = 0$

For the case in which s has children $v_1, \ldots, v_k$, we have

- $F_1(r,s) = \min_{i=1,\ldots,k}\{F_3(s,v_i) + \sum_{j=1,\ldots,k;j\neq i} \min\{F_4(s,v_j), F_5(s,v_j)\}\}$.

  In order to obtain a maximal matching, we need to match s with some of its children, say v_i. Since r will be unmatched, v_i should not have an unmatched neighbor, in order to prevent an augmenting path of length 3. When considering the trees T_{sv_j} for $j \neq i$, the vertex s will have the status of "already matched". Furthermore, since we are already assuming that s has an unmatched neighbor, we do not need to care about the vertices v_j being matched or not.

- $F_2(r,s) = 1 + \sum_{i=1,\ldots,k} \min\{F_4(s,v_i), F_5(s,v_i)\}$.

  We will use the edge rs, and then when considering the trees T_{sv_i} for $i = 1, \ldots, k$, the vertex s will have the status of "already matched". Furthermore, since s may or may not have an unmatched neighbor, we can take the minimum over F_4 and F_5 for each of the trees T_{sv_i}.

- $F_3(r,s) = 1 + \sum_{i=1,\ldots,k} F_4(s,v_i)$.

  This case is similar to the previous one, but now the vertex s cannot have unmatched neighbors, so we will just consider F_4 for each of the trees T_{sv_i}.

- $F_4(r,s) = \min\{\min_{i=1,\ldots,k}\{F_2(s,v_i) + \sum_{j=1,\ldots,k;j\neq i} F_4(s,v_j)\}, \min_{i=1,\ldots,k}\{F_3(s,v_i) + \sum_{j=1,\ldots,k;j\neq i} \min\{F_4(s,v_j), F_5(s,v_j)\}\}\}$

  As in the first case, we need to match s with some of its children, say v_i. But now, since r is assumed to be matched, s may or may not have an unmatched neighbor, depending on the matching status of the vertices v_j with $j \neq i$. So we will take the minimum among allowing v_i to have an unmatched neighbor and forcing v_j, $j \neq i$, to be matched, or forbidding v_i to have an unmatched neighbor and allowing v_j, $j \neq i$, to be either matched or not.

- $F_5(r,s) = \sum_{i=1,\ldots,k} F_1(s,v_i)$

  This last case is quite clear.

In this way, in order to obtain the cardinality of a minimum strongly maximal matching of a nontrivial tree T, we can root it at a leaf r whose neighbor is s and compute $\min\{F_1(r,s), F_2(r,s)\}$. By keeping some extra information, we can also obtain in polynomial time the matching itself. $\square$

5.2 Tree-cographs

A graph is a *tree-cograph* if it can be constructed from trees by disjoint union and complement operations. Tree-cographs have been introduced by Tinhofer [14] as a generalization of trees and cographs.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \emptyset$. The *union* of G_1 and G_2 is the graph $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$, and the *join* of G_1 and G_2 is the graph $G_1 \vee G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup V_1 \times V_2)$. Note that $\overline{G_1 \vee G_2} = \overline{G_1} \cup \overline{G_2}$.

Tree-cographs can be recursively defined as follows: a graph G is a tree-cograph if and only if

(i) G is a tree or a co-tree, or

(ii) G is the union of two tree-cographs G_1 and G_2, or

(iii) G is the join of two tree-cographs G_1 and G_2.

Notice that if (i) in the above definition is replaced by "G is a single vertex" then, the obtained graph is a cograph.

The notion of *dominance sequence* has been introduced in [3] in order to compute the b-chromatic number of P_4-sparse graphs and, in particular, cographs. Formally, given a graph G, the dominance sequence $\mathrm{dom}_G \in \mathbb{Z}^{\mathbb{N}_{\geq \chi(G)}}$, is defined such that $\mathrm{dom}_G[t]$ is the maximum number of distinct color classes admitting dominant vertices in any coloring of G with t colors, for every $t \geq \chi(G)$. Note that it suffices to consider this sequence until $t = |V(G)|$, since $\mathrm{dom}_G[t] = 0$ for $t > |V(G)|$. Therefore, in the sequel we shall consider only the *dominance vector* $(\mathrm{dom}_G[\chi(G)], \ldots, \mathrm{dom}_G[|V(G)|])$. Notice that a graph G admits a b-coloring with t colors if and only if $\mathrm{dom}_G[t] = t$. Moreover, it is clear that $\mathrm{dom}_G[\chi(G)] = \chi(G)$.

The following results given in [3] are very important in order to compute the b-chromatic number of graphs that can be decomposed recursively in modules via disjoint union or join operations.

Theorem 10 ([3])**.** *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. If $G = G_1 \cup G_2$ and $t \geq \chi(G)$, then*

$$\mathrm{dom}_G[t] = \min\{t, \mathrm{dom}_{G_1}[t] + \mathrm{dom}_{G_2}[t]\}.$$

Theorem 11 ([3])**.** *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. Let $G = G_1 \vee G_2$ and $\chi(G) \leq t \leq |V(G)|$. Let $a = \max\{\chi(G_1), t - |V(G_2)|\}$ and $b = \min\{|V(G_1)|, t - \chi(G_2)\}$. Then $a \leq b$ and*

$$\mathrm{dom}_G[t] = \max_{a \leq j \leq b}\{\mathrm{dom}_{G_1}[j] + \mathrm{dom}_{G_2}[t - j]\}.$$

In order to compute the dominance vector of a tree-cograph and its corresponding b-chromatic number, by Theorems 10 and 11, it is sufficient to compute the dominance vector for both trees and co-trees.

5.2.1 Dominance vector for trees

Irving and Manlove [11] have shown that the b-chromatic number of any tree T is equal to $m(T)-1$ or $m(T)$, depending on the existence of a unique vertex in T called a *pivot*, defined as follows.

A vertex v of T is called *dense* if $d(v) \geq m(T)-1$. Call T *pivoted* if T has exactly $m(T)$ dense vertices, and contains a distinguished vertex v, called a *pivot* of T, such that: (1) v is not dense, (2) each dense vertex is adjacent either to v or to a dense vertex adjacent to v, and (3) every dense vertex adjacent to v and to another dense vertex has degree $m(T)-1$.

Irving and Manlove [11] showed that a pivot is unique when it exists and that pivoted trees can be recognized in linear time. Moreover, they obtain the following result.

Theorem 12 ([11])**.** *Let T be a tree. If T is pivoted then $\chi_b(T) = m(T)-1$, otherwise $\chi_b(T) = m(T)$. In both cases, a b-coloring of T with $\chi_b(T)$ colors can be obtained in linear time.*

It is known that chordal graphs are b-continuous [12, 6] and thus trees are b-continuous as well. Therefore, we may derive the following result concerning the dominance vector for trees.

Lemma 13. *Let T be a nontrivial tree with maximum degree Δ. Then, $dom_T[i] = i$, for $2 \leq i \leq \chi_b(T)$. Moreover, $dom_T[i] = 0$ for any $i > \Delta+1$.*

Moreover, it is not difficult to obtain a b-coloring of a tree T with i colors from one with $i+1$ colors in polynomial time, for $2 \leq i < \chi_b(T)$.

Lemma 14. *Let T be a pivoted tree. Then, $dom_T[m(T)] = m(T)-1$.*

Proof. By Theorem 12, if T is a pivoted tree, then $\chi_b(T) = m(T)-1$, so $\mathrm{dom}_T[m(T)] \leq m(T)-1$. Consider now the following coloring of T with $m(T)$ colors. Give color 1 to the pivot v of T. Since v is not dense, there are at least two dense vertices at distance 2 of v; give color 1 to one of them, say w. Now color the dense vertices using the $m(T)$ different colors and color their neighbors in such a way that the only dense vertex that is not dominant is the common neighbor to v and w. It is easy to extend this coloring to a proper coloring of T with $m(T)$ colors. □

We now show how to compute the values $\mathrm{dom}_T[i]$ and a coloring of T with i colors and $\mathrm{dom}_T[i]$ dominant vertices in linear time, for $m(T) < i \leq \Delta+1$. For this, we need the following definition. Let T be a tree of maximum degree Δ and let i be an integer such that $m(T) < i \leq \Delta+1$. We define $m_i(T)$ as the number of vertices in T of degree at least $i-1$.

It is not difficult to see that for a tree T, $\mathrm{dom}_T[i] \leq m_i(T) < i$, for values of i with $m(T) < i \leq \Delta+1$.

Lemma 15. *Let T be a tree of maximum degree Δ and let i be an integer with $m(T) < i \leq \Delta + 1$. Then, $dom_T[i] = m_i(T)$, and a coloring of T with i colors and $m_i(T)$ dominant vertices can be computed in linear time.*

Proof. For convenience, set $k = m_i(T)$. As $i \leq \Delta + 1$, we have $k > 0$. Let P be a path disjoint from T that contains the $i - k + 3 \geq 4$ vertices $x, y, v_1, v_2, \ldots, v_{i-k}, z$ in this order. We construct a tree T' disjoint from T by taking P and pending $i-3$ leaves from each vertex v_j, with $1 \leq j \leq i-k$. Obtain T'' from T and T' by adding an edge between x and some leaf h of T.

By construction, $m_i(T'') = i$, and thus also $m(T'') = i$. Further, T' contains the dense vertices $v_1, v_2, \ldots, v_{i-k}$ of T'', and T contains k dense vertices of T''. So, as both x and y have degree 2 in T'' (and thus either both or none of them are dense), we see that T'' is not pivoted. Hence, Theorem 12 yields that $\chi_b(T'') = m(T'') = i$, and a b-coloring of T'' with i colors can be computed in linear time.

Now, the dominant vertices in T'' are exactly the k vertices of degree at least $i - 1$ in T and the $i - k$ vertices $v_1, \ldots, v_{i-k}$ in T'. Therefore, by removing the tree T' from T'' we obtain the desired coloring of T with i colors and exactly k dominant vertices. Moreover, notice that the distance in T'' between a dense vertex in T and a dense vertex in T' is at least equal to 4. Hence, by using Irving's and Manlove's algorithm [11] for b-coloring T'' with i colors, we can forget the tree T' and thus, the coloring of T with i colors and $m_i(T)$ dominant vertices can be done in $O(|V(T)|)$ time. □

5.2.2 Dominance vector for co-trees

Let G be a graph and M be a matching of it. Let $S_1(G, M)$ be the number of unmatched vertices that have at least an unmatched neighbor and $S_2(G, M)$ be the number of edges of M that are the center of an augmenting path of length 3 for M. Now, let $F(G, k)$ be the minimum of $S_1(G, M) + S_2(G, M)$ over all the matchings M of G with $|M| = k$.

Now, let G be a graph with stability at most two and consider a coloring of it. Let M be the matching of $\overline{G}$ corresponding to that coloring. The number of color classes without a dominant vertex is exactly $S_1(\overline{G}, M) + S_2(\overline{G}, M)$. So, for $\chi(G) \leq i \leq |V(G)|$, $\mathrm{dom}_G[i] = i - F(\overline{G}, |V(G)| - i)$. We will show how to compute $F(T, k)$ for a tree T and a nonnegative integer k in polynomial time.

Theorem 16. *If G is a co-tree, then dom_G can be computed in polynomial time.*

Proof. As we noticed above, if G is nontrivial, then the problem is equivalent to compute $F(\overline{G}, k)$, for $\chi(G) \leq k \leq |V(G)|$. We will do it by dynamic programming. In order to do so, and in a similar way as in Theorem 9, we will define seven functions $F_i(r, s, k)$, $i = 1, \ldots, 7$, for a nontrivial tree

T_{rs} rooted at a leaf r with neighbor s and a nonnegative integer k. As we will apply them to the subtrees of a tree, we will assume that r can have neighbors outside T_{rs}. Nevertheless, we will count for S_2 just the edges of $M \cap E(T_{rs})$ and for S_1 the vertices of $V(T_{rs})$, with the exception of r when it is unmatched but has already an unmatched neighbor outside T_{rs}, in order to avoid double counting.

For $i = 1, \ldots, 7$, $F_i(r, s, k)$ will be the minimum of $S_1(T_{rs}, M) + S_2(T_{rs}, M)$ over all the matchings M with $|M| = k$ such that:

- $F_1(r, s, k)$: r is unmatched and s is matched by M with some vertex different from r.
- $F_2(r, s, k)$: M uses the edge rs and r has no unmatched neighbor outside T_{rs}.
- $F_3(r, s, k)$: M uses the edge rs and r has an unmatched neighbor outside T_{rs}.
- $F_4(r, s, k)$: the vertex s is matched by M with some vertex different from r and the vertex r is already matched with a vertex outside T_{rs}.
- $F_5(r, s, k)$: the vertex s remains unmatched and the vertex r is already matched with a vertex outside T_{rs}.
- $F_6(r, s, k)$: r is unmatched, s remains unmatched, and r has no unmatched neighbor outside T_{rs}.
- $F_7(r, s, k)$: r is unmatched, s remains unmatched, and r has an unmatched neighbor outside T_{rs} (we will not count r for S_1 as we assume it is already counted).

In any case, the value will be ∞ if no such M does exist.

With these definitions, for the base case in which $V(T_{rs}) = \{r, s\}$, we have

- $F_1(r, s, k) = \infty$ (s has no further neighbors)
- For $i = 2, 3$, $F_i(r, s, 1) = 0$ (we define $M = \{rs\}$), $F_i(r, s, k) = \infty$ for $k \neq 1$.
- $F_4(r, s, k) = \infty$ (it is not feasible because s has no further neighbors)
- $F_5(r, s, 0) = 0$, $F_5(r, s, k) = \infty$ for $k \neq 0$.
- $F_6(r, s, 0) = 2$, $F_6(r, s, k) = \infty$ for $k \neq 0$.
- $F_7(r, s, 0) = 1$, $F_7(r, s, k) = \infty$ for $k \neq 0$.

For the case in which s has children $v_1, \ldots, v_\ell$, we have

- $F_1(r,s,0) = \infty$, $F_1(r,s,k) = \min_{k_1+\cdots+k_\ell=k} \min_{i=1,\ldots,\ell} \{F_3(s,v_i,k_i) + \sum_{j=1,\ldots,\ell;j\neq i} \min\{F_4(s,v_j,k_j), F_5(s,v_j,k_j)\}\}$ for $k>0$.

  We need to match s with some of its children, say v_i. Since r will be unmatched, s will have an unmatched neighbor outside T_{sv_i}, and we do not need to distinguish about the vertices v_j being matched or not in T_{sv_j}, for $j \neq i$. When considering the trees T_{sv_j} for $j \neq i$, the vertex s will have the status of "already matched".

- $F_2(r,s,0) = \infty$, $F_2(r,s,k) = \min_{k_1+\cdots+k_\ell=k-1} \sum_{i=1,\ldots,\ell} \min\{F_4(s,v_i,k_i), F_5(s,v_i,k_i)\}$ for $k>0$.

  We will use the edge rs, and then when considering the trees T_{sv_i} for $i = 1,\ldots,\ell$, the vertex s will have the status of "already matched" and we will use $k-1$ edges in total (thus for $k=0$ it is not feasible). Furthermore, since r has no unmatched neighbor, we can take the minimum over F_4 and F_5 for each of the trees T_{sv_i}, and in none of the cases the edge rs will be the center of an augmenting path of length 3.

- $F_3(r,s,0) = \infty$, $F_3(r,s,k) = \min_{k_1+\cdots+k_\ell=k-1} \min\{\sum_{i=1,\ldots,\ell} F_4(s,v_i,k_i), 1 + \sum_{i=1,\ldots,\ell} \min\{F_4(s,v_i,k_i), F_5(s,v_i,k_i)\}\}$ for $k>0$.

  This case is similar to the previous one, but now, since r has an unmatched neighbor, we distinguish between the case in which we consider F_4 for each of the trees T_{sv_i} so that the edge rs will not be the center of an augmenting path of length 3, and the case in which we take the minimum over F_4 and F_5 for each of the trees T_{sv_i} and we allow the edge rs being the center of an augmenting path of length 3. In that case we will assume indeed that the edge rs becomes the center of an augmenting path of length 3, because otherwise the minimum will be attained by the previous case.

- $F_4(r,s,0) = \infty$, $F_4(r,s,k) = \min_{k_1+\cdots+k_\ell=k} \min\{$
  $\min_{i=1,\ldots,\ell}\{F_2(s,v_i,k_i) + \sum_{j=1,\ldots,\ell;j\neq i} F_4(s,v_j,k_j)\}$,
  $\min_{i=1,\ldots,\ell}\{F_3(s,v_i,k_i) + \sum_{j=1,\ldots,\ell;j\neq i} \min\{F_4(s,v_j,k_j), F_5(s,v_j,k_j)\}\}\}$
  for $k>0$.

  As in the first case, we need to match s with some of its children, say v_i. But now, since r is assumed to be matched, s may or may not have an unmatched neighbor, depending on the matching status of the vertices v_j with $j \neq i$. So we will take the minimum among allowing v_i to have an unmatched neighbor and forcing v_j, $j \neq i$, to be matched, or forbidding v_i to have an unmatched neighbor and allowing v_j, $j \neq i$, to be either matched or not.

- $F_5(r,s,k) = \min_{k_1+\cdots+k_\ell=k} \min\{\sum_{i=1,\ldots,\ell} F_1(s,v_i,k_i),$
  $\min_{i=1,\ldots,\ell} \{F_6(s,v_i,k_i) + \sum_{j=1,\ldots,\ell;j\neq i} F_1(s,v_j,k_j)\}$,

$1 + \sum_{i=1,\ldots,\ell} \min\{F_1(s, v_i, k_i), F_7(s, v_i, k_i)\}\}$

We will take the minimum over three cases: either all the v_i will be matched, or exactly one of them will be unmatched, or at least two of them will be unmatched. In this last case we count s as an unmatched vertex with unmatched neighbor but we do not force explicitly two of the v_i to be unmatched, because otherwise the minimum will be attained by one of the previous cases.

- $F_6(r, s, k) = 2 + \min_{k_1+\cdots+k_\ell=k} \sum_{i=1,\ldots,\ell} \min\{F_1(s, v_i, k_i), F_7(s, v_i, k_i)\}$

  We are counting r and s as unmatched vertices with an unmatched neighbor.

- $F_7(r, s, k) = 1 + \min_{k_1+\cdots+k_\ell=k} \sum_{i=1,\ldots,\ell} \min\{F_1(s, v_i, k_i), F_7(s, v_i, k_i)\}$

  We are counting just s as an unmatched vertex with unmatched neighbor, since r is assumed to be already counted.

Notice that as the values of the functions F_i are bounded by the number of vertices of the corresponding tree, and k is also bounded by that number, taking the minimum over $k_1 + \cdots + k_\ell = k$ of some combination of these F_i is equivalent to solve a polynomially bounded number of knapsack problems where both the weights and the utilities are polynomially bounded as well, so this can be done by dynamic programming in polynomial time [4]. We can rewrite all the expressions for $F_1, \ldots, F_7$ as the minimum over a polynomial number of knapsack problems based on the precomputed functions for the corresponding subtrees of the tree.

In this way, in order to obtain $F(T, k)$ for a nontrivial tree T, we can root it at a leaf r whose neighbor is s and compute $\min\{F_1(r, s, k), F_2(r, s, k), F_6(r, s, k)\}$. By keeping some extra information, we can also obtain in polynomial time the matching itself. □

5.2.3 b-continuity and b-monotonicity of tree-cographs

The following result was proved for union and join of graphs.

Lemma 17. *[3] Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $V_1 \cap V_2 = \emptyset$. If G_1 and G_2 are b-continuous, then $G_1 \cup G_2$ and $G_1 \vee G_2$ are b-continuous.*

As a corollary of the lemma, Theorem 4, and the b-continuity of chordal graphs [12, 6], we have the following result.

Theorem 18. *Tree-cographs are b-continuous.*

Concerning the b-monotonicity, the following results are known for general graphs and for union and join of graphs.

Lemma 19. *[3] Let G be a graph. The maximum value of* $\mathrm{dom}_G[t]$ *is attained in* $t = \chi_b(G)$.

Lemma 20. *[3] Let* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *be two graphs such that* $V_1 \cap V_2 = \emptyset$*, and let* $G = G_1 \cup G_2$*. Assume that for every* $t \geq \chi(G_i)$ *and every induced subgraph* H *of* G_i *we have* $\mathrm{dom}_H[t] \leq \mathrm{dom}_{G_i}[t]$*, for* $i = 1, 2$*. Then, for every* $t \geq \chi(G)$ *and every induced subgraph* H *of* G*,* $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ *holds.*

Lemma 21. *[3] Let* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *be two b-continuous graphs such that* $V_1 \cap V_2 = \emptyset$*, and let* $G = G_1 \vee G_2$*. Assume that for every* $t \geq \chi(G_i)$ *and for every induced subgraph* H *of* G_i *we have* $\mathrm{dom}_H[t] \leq \mathrm{dom}_{G_i}[t]$*, for* $i = 1, 2$*. Then, for every* $t \geq \chi(G)$ *and for every induced subgraph* H *of* G*,* $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ *holds.*

In order to prove the b-monotonicity of tree-cographs, we need the following two lemmas.

Lemma 22. *Let* T *be a tree and* H *an induced subgraph of* T*. Then for every* $t \geq 2$*,* $\mathrm{dom}_H[t] \leq \mathrm{dom}_T[t]$.

Proof. It is clear that it holds for $t \leq \chi_b(T)$. If T is a pivoted tree, then either $m(H) < m(T)$ or the connected component of H containing the dense vertices is a pivoted tree as well. In any case, $\mathrm{dom}_H[m(T)] \leq \mathrm{dom}_T[m(T)]$. For $t > m(T)$, let T_j, $j = 1, \ldots, k$, be the connected components of H. It is clear that $\sum_{j=1,\ldots,k} m_t(T_j) \leq m_t(T)$, so by Theorem 10, $\mathrm{dom}_H[t] \leq \mathrm{dom}_T[t]$. □

Lemma 23. *Let* G *be a graph with stability at most two and* H *an induced subgraph of* G*. Then for every* $t \geq \chi(G)$*,* $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$.

Proof. The class of graphs of stability at most two is closed under taking induced subgraphs. Thus we only have to prove that $\mathrm{dom}_G[t]$ for a fixed t is monotonously decreasing under the deletion of a vertex. Let $H = G - v$ for some vertex v of G. It is clear that $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ for $\chi(G) \leq t \leq \chi_b(G)$ and for $t > |V(H)| = |V(G)| - 1$. For $|V(H)| \geq t > \chi_b(G)$, as we observed before, since G and H have stability at most two, $\mathrm{dom}_G[t] = t - F(\overline{G}, |V(G)| - t)$, and $\mathrm{dom}_H[t] = t - F(\overline{H}, |V(H)| - t)$, where $F(X, k)$ stands for the minimum sum of the number of unmatched vertices that have at least an unmatched neighbor and the number of edges of M that are the center of an augmenting path of length 3 for M, over all the matchings M of a graph X with $|M| = k$. Then $\mathrm{dom}_G[t] \geq \mathrm{dom}_H[t]$ if and only if $F(\overline{G}, |V(G)| - t) = F(\overline{G}, |V(H)| + 1 - t) \leq F(\overline{H}, |V(H)| - t)$.

Let M be a matching of $\overline{H}$ that realizes this minimum, and consider M as a matching of $\overline{G}$. We need to find a matching M' of $\overline{G}$ with $|M'| = |M| + 1$, which is always posible, since $t > \chi(G)$ and then $|V(G)| - t$ is strictly smaller than the size of a maximum matching of $\overline{G}$.

We will consider now three cases. If v has an unmatched neighbor w, then let $M' = M \cup \{vw\}$. In this way, $S_1(\overline{G}, M') \leq S_1(\overline{H}, M)$ and no edge of M becomes the center of an augmenting path of length 3 for M' in $\overline{G}$. Moreover, if vw is the center of an augmenting path of length 3 for M' in $\overline{G}$, then w was for M an unmatched vertex of $\overline{H}$ having an unmatched neighbor. In this case, $S_2(\overline{G}, M') = S_2(\overline{H}, M) + 1$ but $S_1(\overline{G}, M') \leq S_1(\overline{H}, M) - 1$. In any case, $F(\overline{G}, |V(G)| - t) \leq S_1(\overline{G}, M') + S_2(\overline{G}, M') \leq S_1(\overline{H}, M) + S_2(\overline{H}, M) = F(\overline{H}, |V(H)| - t)$.

If v has no unmatched neighbor but it is the end of an augmenting path of length 3 for M in $\overline{G}$, say $vxyw$, let $M' = M \setminus \{xy\} \cup \{vx, yw\}$. There are no new unmatched vertices, so $S_1(\overline{G}, M') \leq S_1(\overline{H}, M)$ and no edge of M becomes the center of an augmenting path of length 3 for M' in $\overline{G}$. Neither does vx, since v had no unmatched neighbor. If yw is the center of an augmenting path of length 3 for M' in $\overline{G}$, then w was for M an unmatched vertex of $\overline{H}$ having an unmatched neighbor. In this case $S_2(\overline{G}, M') = S_2(\overline{H}, M) + 1$ but $S_1(\overline{G}, M') \leq S_1(\overline{H}, M) - 1$, so we are done.

Finally, if v does not have an unmatched neighbor and it is not the end of an augmenting path of length 3 for M in $\overline{G}$, i.e., $S_1(\overline{G}, M) = S_1(\overline{H}, M)$ and $S_2(\overline{G}, M) = S_2(\overline{H}, M)$, let P be a minimum length augmenting path in $\overline{G}$ with respect to M, and let $M' = (M \setminus E(P)) \cup (E(P) \setminus M)$. There are no new unmatched vertices, so $S_1(\overline{G}, M') \leq S_1(\overline{H}, M)$ and no edge of M becomes the center of an augmenting path of length 3 for M' in $\overline{G}$. If P is of length 1, then $S_2(\overline{G}, M') \leq S_2(\overline{H}, M) + 1$ but $S_1(\overline{G}, M') \leq S_1(\overline{H}, M) - 2$, so we are done. If P is of length 3, we are eliminating an augmenting path of length 3 and no new edge becomes the center of an augmenting path of length 3, because otherwise there were couples of adjacent unmatched vertices and we are supposing P is of minimum length. Thus $S_1(\overline{G}, M') = S_1(\overline{H}, M) = 0$ and $S_2(\overline{G}, M') \leq S_2(\overline{H}, M) - 1$, so we are done. If there were no augmenting paths of length 1 or 3, M is a strongly maximal matching of $\overline{G}$ and then $|V(G)| - |V(H)| + t = t + 1 \leq \chi_b(\overline{G})$, a contradiction because we were supposing $t > \chi_b(\overline{G})$. □

So, we can conclude the following.

Theorem 24. *Tree-cographs are b-monotonic.*

Proof. As tree-cographs are hereditary, it is enough to prove that given a tree-cograph G, $\chi_b(G) \geq \chi_b(H)$, for every induced subgraph H of G. By the decomposition structure of tree-cographs [14] and Theorem 18, Lemmas 20, 21, 22, and 23, an induction argument shows that for every tree-cograph G, every $t \geq \chi(G)$, and every induced subgraph H of G, $\mathrm{dom}_H[t] \leq \mathrm{dom}_G[t]$ holds. Let G be a tree-cograph, and let H be an induced subgraph of G. If $\chi_b(H) < \chi(G)$, then $\chi_b(H) < \chi_b(G)$. Otherwise, $\chi_b(H) = \mathrm{dom}_H[\chi_b(H)] \leq \mathrm{dom}_G[\chi_b(H)]$, and by Lemma 19 $\mathrm{dom}_G[\chi_b(H)] \leq \mathrm{dom}_G[\chi_b(G)] = \chi_b(G)$. Hence $\chi_b(G) \geq \chi_b(H)$. □

Acknowledgement: We would like to thank the anonymous referees for their careful reading and suggestions that helped us to improve the paper.

References

[1] C. Berge. Two theorems in Graph Theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43:842–844, 1957.

[2] H.L. Bodlaender. Achromatic number is NP-complete for cographs and interval graphs. *Information Processing Letters*, 31:135–138, 1989.

[3] F. Bonomo, G. Durán, F. Maffray, J. Marenco, and M. Valencia-Pabon. On the b-coloring of cographs and P_4-sparse graphs. *Graphs and Combinatorics*, 25(2):153–167, 2009.

[4] G.B. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5:266–277, 1957.

[5] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[6] T. Faik. *La b-continuité des b-colorations : complexité, propriétés structurelles et algorithmes.* PhD thesis, L.R.I., Université Paris-Sud, Orsay, France, 2005.

[7] F. Harary and S. Hedetniemi. The achromatic number of a graph. *Journal of Combinatorial Theory*, 8:154–161, 1970.

[8] F. Havet, C. Linhares-Sales, and L. Sampaio. b-coloring of tight graphs. *Discrete Applied Mathematics*, 160(18):2709–2715, 2012.

[9] C. T. Hoàng and M. Kouider. On the b-dominating coloring of graphs. *Discrete Applied Mathematics*, 152:176–186, 2005.

[10] C. T. Hoàng, C. Linhares Sales, and F. Maffray. On minimally b-imperfect graphs. *Discrete Applied Mathematics*, 157(17):3519–3530, 2009.

[11] R. W. Irving and D. F. Manlove. The b-chromatic number of a graph. *Discrete Applied Mathematics*, 91:127–141, 1999.

[12] J. Kára, J. Kratochvíl, and M. Voigt. b-continuity. Technical Report M 14/04, Technical University Ilmenau, Faculty of Mathematics and Natural Sciences, 2004.

[13] J. Kratochvíl, Zs. Tuza, and M. Voigt. On the b-chromatic number of a graph. *Lecture Notes in Computer Science*, 2573:310–320, 2002.

[14] G. Tinhofer. Strong tree-cographs are Birkoff graphs. *Discrete Applied Mathematics*, 22(3):275–288, 1989.

[15] M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.