Constrained Domatic Bipartition on Trees

Giovanni Andreatta, Carla De Francesco, Luigi De Giovanni, Paolo Serafini

Abstract. Given an undirected graph, the Constrained Domatic Bipartition Problem (CDBP) consists in determining a bipartition, if it exists, of the nodes into two dominating sets, with the additional constraint that one of the two subsets has a given cardinality. The problem is NP-hard in general and in this paper we focus on trees. First, we provide explicit solutions in simple cases, i.e., stars and paths. Then, we provide a polyhedral representation for all domatic bipartitions of a tree. Although the matrix associated with the polyhedron is not totally unimodular, we prove that all its vertices have integral components. Adding the cardinality constraint, the resulting polyhedron will generally lose this property. We then propose a constructive, dynamic programming algorithm for CDBP on trees, that is able to simultaneously find a solution for all possible cardinalities. The proposed algorithm is polynomial with complexity $O(n^3)$, where n is the number of nodes. Finally, we discuss the extension of CDBP to the weighted case, show that it is NP-hard and provide a pseudo-polynomial algorithm for the problem.

1 Introduction

Given an undirected graph $G = (V, E)$, a subset of nodes is *dominating* if each node of G is either in the subset or is adjacent to some node in the subset. A *domatic partition* is a partition of V into dominating sets.

The problem of splitting V into 2 dominating sets has always a positive answer, assuming G has no isolated nodes [Ore, 1962]. In fact, if the graph is a tree then it is connected and bipartite and the two subsets of nodes in the bipartition give two dominating sets [Cockayne and Hedetniemi, 1977]. Otherwise, it suffices to consider a spanning tree for each connected component.

By constraining the cardinalities of the two sets we have the following problem, that we call Constrained Domatic Bipartition Problem (CDBP): given a graph $G = (V, E)$ and a number p with $1 \leq p < |V|$, determine a bipartition, if it exists, of V into 2 dominating sets V' and V'' such that $|V'| = p$.

This problem has been brought to our attention while working on a problem of splitting the node set of a graph into subsets such that, roughly speaking, all subsets are as close as possible to each other. We fully investigated this problem in Andreatta et al. [2015], but the computational complexity of the case with two balanced sets of nodes was left as an open question. This case is closely connected to CDBP and, to the best of our knowledge, the computational complexity of CDBP has not been investigated in the literature.

In a separate paper [Andreatta et al., 2016] we show that CDBP is NP-hard in general and in this paper we show that it is polynomial on trees by providing a dynamic programming algorithm which produces a bipartition for any value of p, if it exists.

We also investigate CDBP from a polyhedral point of view. We show that the domatic bipartition polyhedron (i.e., the polyhedron whose vertices are incidence vectors of one of the two dominating sets) can be represented by a set of $O(|V|)$ linear inequalities in $O(|V|)$ variables. In fact, the vertices associated to this representation have binary components, even if the inequalities do not correspond to a totally unimodular matrix. This allows to find a domatic bipartition with minimum p by solving a linear programming problem. It is already known that finding a domatic bipartition on a tree with the minimum value of p can be done in linear time by using the algorithm proposed in Cockayne and Hedetniemi [1975, 1977]. However, adding a cardinality constraint may introduce fractional vertices and therefore linear programming cannot be used to solve CDBP. The polynomial algorithm mentioned above can be used in this case.

The paper is structured as follows. In Section 2 we recall the H-Domatic Partition problem, the H-Shift Coloring problem and the m-node Domatic Partition problem. We discuss some similarities with CDBP and highlight the important differences. In Section 3 we preliminarily consider special classes of trees, namely, stars and paths, for which the solution is straightforward, and simple necessary and sufficient conditions can be given on p for CDBP to have a solution. In Section 4 we provide a polyhedral representation of all domatic bipartitions of a tree. The dynamic programming algorithm to solve CDBP is described in Section 5. Finally, Section 6 discusses the extension of CDBP to the weighted case, shows that it is NP-hard on trees and provides a pseudo-polynomial algorithm for the problem.

2 Connections with existing literature

Given G and an integer H, the H-Domatic Partition Problem asks whether there exists a partition of V into H dominating subsets. The optimization version of the problem gives rise to the Domatic Number Problem, where the maximum H (called *domatic number*) has to be found, and the Domatic Partition Problem, where a domatic partition having a maximum H has to be determined. Problems related to the domatic partition of a graph have been the object of several studies, showing, among other results, that the H-Domatic Partition Problem is NP-complete for $H \geq 3$ Garey and Johnson [1979], while determining the domatic number is polynomial on some special classes of graphs (Cockayne and Hedetniemi [1977]; Manacher and Mankus [1996]; Poon et al. [2012] among others). The H-Domatic Partition Problem with $H = 2$ has always a positive answer, assuming G has no isolated nodes [Ore, 1962].

The H-Domatic Partition Problem has also relations with the H-Shift Coloring Problem [Andreatta et al., 2015]: given a graph $G = (V, E)$ (with edges of arbitrary length and no weights on the nodes) and an integer H, we want to determine a partition of the nodes into H subsets $V_1, V_2, \ldots, V_H$ such that the total distance $D_{tot} = \sum_{v \in V} \sum_{i=1}^{H} d(v; V_i)$ is minimized, where $d(v; U) = \min \{d(v; u) : u \in U\}$ and $d(v; u)$ is the length of a shortest path between v and u. The problem is NP-hard in general [Andreatta et al., 2015] and polynomial on trees [Andreatta et al., 2014]. When all distances are unitary, then an optimal H-shift coloring is an H-domatic partition if and only if $D_{tot} = |V|(H - 1)$.

A problem related to CDBP is the m-node domatic partition problem Chang [1994], where one has to find the maximum number of pairwise disjoint dominating sets whose union has cardinality at most m. Since the maximum number of pairwise disjoint dominating sets in a tree is two, solving the m-node domatic partition problem on a tree ends up with finding conditions on m under which a domatic bipartition exists [Poon et al., 2012]. However, no explicit constraints on the cardinality of the sets in this partition is taken into account, as required by CDBP.

To the best of our knowledge, the only work concerning CDBP is Andreatta et al. [2016], where it is shown that the addition of the cardinality constraint makes the problem NP-hard. In this paper we will focus on trees, showing that in this case CDBP is polynomial.

3 Subclasses of trees

In the following, we solve CDBP by searching for a 2-coloring of the nodes, let them be b (black) and w (white), such that each node is adjacent to a node of different color and the cardinality of the set of black nodes is equal to p. Since we will focus on trees, we will use the notation $T = (V, E)$ instead of G. Let n be the cardinality of set V.

Proposition 1 *If T is a star, then CDBP has a solution only for $p = 1$ or $p = n - 1$.*

Proof. One dominating set is the center of the star, the other dominating set contains all the leaves.

Proposition 1 shows that the set of values p for which CDBP has a solution is not necessarily an interval of integers.

Proposition 2 *If T is a path, then CDBP has a solution if and only if $n/3 \leq p \leq 2\,n/3$.*

Proof.
$\Rightarrow$) If the problem has a solution, then it is necessary that $p \geq n/3$, since any black node can be adjacent to at most two white nodes. Analogously, it is also necessary that $n - p \geq n/3$, that is $p \leq 2\,n/3$.
$\Leftarrow$) Color the path by concatenating s_b strings bwb, s_w strings wbw and s_0 strings bw. No matter how we concatenate the strings, the coloring of the path is a domatic bipartition. If $n/3 \leq p \leq n/2$ we put $s_b = 0$, $s_w = n - 2\,p$ and $s_0 = 3\,p - n$. If $n/2 \leq p \leq 2\,n/3$ we put $s_b = 2\,p - n$, $s_w = 0$ and $s_0 = 2\,n - 3\,p$.

4 The Domatic Bipartition Polytope

CDBP can be formulated as an Integer Linear Programming (ILP) model. Let us consider a simple (i.e., without multiple edges and loops) connected graph $G = (V, E)$, not necessarily a tree, and without loss of generality, assume $n = |V| \geq 2$. We associate a binary decision variable x_i to each node i of V, with $x_i = 1$ if i is colored black and 0 otherwise.

Let $S(i)$ be the star in G centered in i (including i itself) and $d(i)$ the degree of i in G. $S(i)$ is also called in the literature the *closed neighborhood* of i and denoted by $N[i]$. Given an integer p, the following ILP formulation represents all feasible solutions to CDBP:

$$1 \leq \sum_{j \in S(i)} x_j \leq d(i) \qquad \forall\, i \in V, \tag{1}$$

$$0 \leq x_i \leq 1 \qquad \forall\, i \in V, \tag{2}$$

$$x_i \in Z \qquad \forall\, i \in V, \tag{3}$$

$$\sum_{i \in V} x_i = p. \tag{4}$$

The left inequalities in (1) state that, for every star, at least one node must be black; the right inequalities in (1) state that not all nodes of a star can be black. We will call the inequalities (1) *star inequalities*. Inequalities (2) and (3) state that the vector $x = (x_1, x_2, \ldots, x_n)$ is binary, and (4) is the cardinality constraint.

Let $P(G) \in R^n$ be the polytope associated to inequalities (1) and (2). Notice that $P(G)$ does not consider the cardinality constraint. The following properties of $P(G)$ are easy to check:

- If $x \in P(G)$, then also its complement $\mathbf{1} - x \in P(G)$;
- $P(G)$ is not empty: in fact, the point $(1/2, 1/2, \ldots, 1/2)$ satisfies all inequalities (1) and (2);
- integral solutions, i.e., domatic bipartitions, are vertices of $P(G)$.

The system of star inequalities (1) can be reformulated in matrix form as:

$$\mathbf{1} \leq Sx \leq d \tag{5}$$

with obvious meaning for S and d. Notice that $S = A + I$ where A is the adjacency matrix of G and I the identity matrix. Hence S is symmetric. We are going to prove that, when G is a tree T, every vertex of

the polyhedron $P(T)$ is integral and it corresponds to a domatic bipartition, so that $P(T)$ is the domatic bipartition polyhedron. Note that the integrality of the vertices holds even if the matrix S is not totally unimodular (as, e.g., for a star with three leaves).

Lemma 3 *Given a tree T, if i is a node adjacent to a leaf, then the pair of star inequalities $1 \leq \sum_{j \in S(i)} x_j \leq d(i)$ is redundant.*

Proof. Let k be a leaf adjacent to i. The star inequalities for node k are $1 \leq x_i + x_k \leq d(k) = 1$ so that $x_i + x_k = 1$.
From $1 = x_i + x_k \leq x_i + x_k + \sum_{j \in S(i) \setminus \{k,i\}} x_j = \sum_{j \in S(i)} x_j$, we obtain that the first star inequality for node i is automatically satisfied.
Furthermore, the other star inequality for i derives from

$$\sum_{j \in S(i)} x_j = x_i + x_k + \sum_{j \in S(i) \setminus \{k,i\}} x_j = 1 + \sum_{j \in S(i) \setminus \{k,i\}} x_j \leq 1 + \sum_{j \in S(i) \setminus \{k,i\}} 1 = 1 + d(i) - 1 = d(i).$$

Notice that Lemma 3 holds also if i itself is a leaf (i.e., $n = 2$). In this case, the system of star inequalities (5) is equivalent to $x_i + x_j = 1$.

Lemma 4 *Given any tree $T = (V, E)$ with $|V| \geq 3$, at least one of the following two cases is satisfied:*
Case a: there is a node adjacent to at least two leaves;
Case b: there is a node of degree two adjacent to a leaf.

Proof. Choose any node r as root of the tree and choose a leaf z having maximum distance (i.e., w.r.t. the number of edges) from r. If the parent of z has degree two then case b is verified. Otherwise, the parent of z has at least one other descendant that is also a leaf, given that z has maximum distance from r.

Theorem 5 *Given a tree T, the vertices of $P(T)$ are integer.*

Proof. We will prove that any fractional point of $P(T)$ cannot be a vertex of $P(T)$. The proof is by induction on $n = |V|$. The assertion is obviously true for $n = 2$ and for $n = 3$.
So assume that the assertion is true for all trees having up to n nodes, for a fixed n.
Consider a tree T_{n+1} with $n + 1$ nodes, $n \geq 3$. Let x be a fractional solution of $P(T_{n+1})$ and, by Lemma 4, let us distinguish the following two cases:

Case a : there is a node adjacent to at least two leaves,
Case b : there is a node of degree two adjacent to a leaf.

In case a we denote by v the node adjacent to the two leaves denoted u and w. In case b we denote by v the node of degree two adjacent to the leaf denoted w and to the node denoted u.
In both cases T_n is the tree obtained by removing the node w and the edge (v, w). Let y be the restriction of x to T_n. We observe that a fractional solution x of $P(T_{n+1})$ cannot have x_w as the only fractional component, because a fractional value x_w implies a fractional value of $x_v = 1 - x_w$ (from the star constraints related to node w). Hence if x is fractional, y is fractional as well.
The basic idea of the proof is that, given a fractional point x of $P(T_{n+1})$, we consider its restriction y to T_n. If it happens that y is feasible for $P(T_n)$ then, by the inductive assumption, y is not a vertex and can be written as a convex combination of two feasible solutions y' and y'' of $P(T_n)$. Then we have to lift y' and y'' to R^{n+1} to find a convex combination for x in $P(T_{n+1})$ and thus establishing that x cannot be a vertex.

However, the procedure is not as straightforward as it may seem. The restriction y is not necessarily feasible for $P(T_n)$ and we need some tricks to modify y into a feasible fractional point. Then we have to find out a way to lift y' and y'' to feasible points x' and x'' in R^{n+1}. The following proof works out all these details.

Proof in case a.

In this case y is obviously a fractional solution of $P(T_n)$, since only star inequalities for node v should be checked, but they are redundant, see Lemma 3. By the inductive hypothesis there must exist a vector $\delta \in R^n$ such that:

$$\delta \neq \mathbf{0}, \quad y' = y + \delta \in P(T_n), \quad y'' = y - \delta \in P(T_n), \quad y = (y' + y'')/2.$$

Let $x' \in R^{n+1}$ and $x'' \in R^{n+1}$ be defined as

$$x'_i = \begin{cases} y'_i & \text{if } \ i \neq w, \\ 1 - y'_v & \text{if } \ i = w \end{cases} \qquad \text{and} \qquad x''_i = \begin{cases} y''_i & \text{if } \ i \neq w, \\ 1 - y''_v & \text{if } \ i = w. \end{cases}$$

It is easy to check that both x' and x'' belong to $P(T_{n+1})$, and that $x = (x' + x'')/2$. This shows that, in case a, if x is a fractional solution of $P(T_{n+1})$ then x cannot be a vertex.

Proof in case b.

Given a fractional solution x of $P(T_{n+1})$, let us distinguish three more cases:

Case b.1 : $x_v = 1 - x_u$,
Case b.2 : $x_v \neq 1 - x_u$ and $0 < x_v < 1$,
Case b.3 : $x_v \neq 1 - x_u$ and x_v integer.

Proof in case b.1.

In this case y is obviously a fractional solution of $P(T_n)$, since $x_u + x_v = 1$, and the proof goes as in case a.

Proof in case b.2.

First of all, it is possible that the only fractional components of x are x_v and x_w. If this is the case, then notice that we must have

$$1 < \sum_{j \in S(u)} x_j < d(u),$$

so that defining $x' \in R^{n+1}$ and $x'' \in R^{n+1}$ as

$$x'_i = \begin{cases} x_i & \text{if } \ i \neq v, w, \\ x_i + \epsilon & \text{if } \ i = v, \\ x_i - \epsilon & \text{if } \ i = w \end{cases} \qquad \text{and} \qquad x''_i = \begin{cases} x_i & \text{if } \ i \neq v, w, \\ x_i - \epsilon & \text{if } \ i = v, \\ x_i + \epsilon & \text{if } \ i = w, \end{cases}$$

where $0 < \epsilon < \min\{x_v, x_w\}$, we have that both x' and x'' belong to $P(T_{n+1})$, and that $x = (x' + x'')/2$. In this case x cannot be a vertex.

Thus, we may assume that at least one of the components x_i is fractional, for $i \neq v, w$. In this case, we need to modify the definition of $y \in R^n$. Let

$$y_i = \begin{cases} x_i & \text{if } \ i \neq v, \\ 1 - x_u & \text{if } \ i = v. \end{cases}$$

We have that $y \in P(T_n)$: the only inequalities to be checked are the star inequalities for u, which are redundant (see Lemma 3). Since y is a fractional solution of $P(T_n)$, by the inductive hypothesis there must exist a vector $\delta \in R^n$ such that:

$$\delta \neq \mathbf{0}, \quad y' = y + \delta \in P(T_n), \quad y'' = y - \delta \in P(T_n), \quad y = (y' + y'')/2.$$

Notice that, for any value of γ (with $0 < \gamma \leq 1$), letting

$$y'(\gamma) = y + \gamma\,\delta \quad \text{and} \quad y''(\gamma) = y - \gamma\,\delta,$$

we continue to have $y'(\gamma) \in P(T_n)$, $y''(\gamma) \in P(T_n)$ and $y = (y'(\gamma) + y''(\gamma))/2$.
Define $\Delta = \sum_{i \in S(u)\setminus v} \delta_i$. Furthermore, let $\bar{\gamma}$ be any number such that

$$0 < \bar{\gamma} < \min\left\{ \frac{\min\{x_v, 1 - x_v\}}{|\Delta|}\ ,\ 1 \right\}$$

if $\Delta \neq 0$, and $\bar{\gamma} = 1$ if $\Delta = 0$. Let $x' \in R^{n+1}$ and $x'' \in R^{n+1}$ be defined as

$$x'_i = \begin{cases} y'_i(\bar{\gamma}) & \text{if } \ i \neq v, w, \\ x_v - \bar{\gamma}\Delta & \text{if } \ i = v, \\ 1 - x'_v & \text{if } \ i = w \end{cases} \qquad \text{and} \qquad x''_i = \begin{cases} y''_i(\bar{\gamma}) & \text{if } \ i \neq v, w, \\ x_v + \bar{\gamma}\Delta & \text{if } \ i = v, \\ 1 - x''_v & \text{if } \ i = w. \end{cases}$$

We have that $x = (x' + x'')/2$. In order to verify that x' belongs to $P(T_{n+1})$, we have to verify that the inequalities (1) and (2) are satisfied. The only non trivial checks concern the proof that $0 \leq x'_v \leq 1$, which is true thanks to the choice of $\bar{\gamma}$, and $1 \leq \sum_{j \in S(u)} x'_j \leq d(u)$, which is true because

$$\begin{aligned} \sum_{j \in S(u)} x'_j &= \sum_{j \in S(u)\setminus\{v\}} x'_j + x'_v = \sum_{j \in S(u)\setminus\{v\}} y'_j(\bar{\gamma}) + x'_v \\ &= \sum_{j \in S(u)\setminus\{v\}} (y_j + \bar{\gamma}\,\delta_j) + (x_v - \bar{\gamma}\,\Delta) \\ &= \sum_{j \in S(u)\setminus\{v\}} (x_j + \bar{\gamma}\,\delta_j) + (x_v - \bar{\gamma}\,\Delta) \\ &= \sum_{j \in S(u)} x_j + \bar{\gamma} \sum_{j \in S(u)\setminus\{v\}} \delta_j - \bar{\gamma}\,\Delta = \sum_{j \in S(u)} x_j\ . \end{aligned}$$

Analogously we can verify that $x'' \in P(T_{n+1})$.

Proof in case b.3.

Since $x_v \neq 1 - x_u$ and x_v integer, without loss of generality we may assume that $x_v > 1 - x_u$ (the case $x_v < 1 - x_u$ is symmetric) so that $x_v = 1$ and $x_u > 0$.
Let us distinguish three further cases:

Case b.3.1 : x_u is also integer,
Case b.3.2 : $0 < x_u < 1$ and $\sum_{j \in S(u)} x_j = d(u)$,
Case b.3.3 : $0 < x_u < 1$ and $\sum_{j \in S(u)} x_j < d(u)$.

Proof in case b.3.1.

If x_u is integer then $x_u = 1$. Let T_{n-1} be the tree obtained from T_{n+1} by eliminating nodes v, w and edges (u, v) and (v, w). Let y be the restriction of x to T_{n-1}. Obviously y is a fractional solution of $P(T_{n-1})$. By the inductive hypothesis there must exist a vector $\delta \in R^{n-1}$ such that:

$$\delta \neq \mathbf{0}, \quad y' = y + \delta \in P(T_{n-1}), \quad y'' = y - \delta \in P(T_{n-1}), \quad y = (y' + y'')/2.$$

Let $x' \in R^{n+1}$ and $x'' \in R^{n+1}$ be defined as

$$x'_i = \begin{cases} y'_i \text{ if } & i \neq v, w, \\ 1 \text{ if } & i = v, \\ 0 \text{ if } & i = w \end{cases} \qquad \text{and} \qquad x''_i = \begin{cases} y''_i \text{ if } & i \neq v, w, \\ 1 \text{ if } & i = v, \\ 0 \text{ if } & i = w. \end{cases}$$

It is easy to check that both x' and x'' belong to $P(T_{n+1})$, and that $x = (x' + x'')/2$.

Proof in case b.3.2.

Let T_{n-1} and y be defined as in case b.3.1. From $\sum_{j \in S(u)} x_j = d(u)$ we have

$$\sum_{j \in S(u) \setminus \{v\}} y_j = d(u) - 1 \geq 1,$$

since u is not a leaf of T_{n+1} in case b. Then y is a fractional solution of $P(T_{n-1})$, and we can define y' and y'' in $P(T_{n-1})$, x' and x'' in $P(T_{n+1})$ as in case b.3.1 and again we obtain that $x = (x' + x'')/2$.

Proof in case b.3.3.

We have that $0 < x_u < 1$ and $1 < \sum_{j \in S(u)} x_j < d(u)$. Notice that in this case if we define y as in case b.3.1, it may not be in $P(T_{n-1})$, since $\sum_{j \in S(u) \setminus \{v\}} x_j$ could be strictly less than 1. Define $y \in R^n$ as

$$y_i = \begin{cases} x_i & \text{if } i \neq v, \\ 1 - x_u & \text{if } i = v. \end{cases}$$

It is easy to check that $y \in P(T_n)$. Since y is a fractional solution of $P(T_n)$, by the inductive hypothesis there must exist a vector $\delta \in R^n$ such that:

$$\delta \neq \mathbf{0}, \quad y' = y + \delta \in P(T_n), \quad y'' = y - \delta \in P(T_n), \quad y = (y' + y'')/2.$$

Notice that, for any value of γ (with $0 < \gamma \leq 1$), letting

$$y'(\gamma) = y + \gamma\,\delta \quad \text{and} \quad y''(\gamma) = y - \gamma\,\delta,$$

we continue to have $y'(\gamma) \in P(T_n)$, $y''(\gamma) \in P(T_n)$ and $y = (y'(\gamma) + y''(\gamma))/2$.
Define $\Delta = \sum_{i \in S(u) \setminus \{v\}} \delta_i$. Furthermore, let $\bar{\gamma}$ be any number such that

$$0 < \bar{\gamma} < \min \left\{ \frac{\min \left\{ \sum_{j \in S(u)} x_j - 1, d(u) - \sum_{j \in S(u)} x_j \right\}}{|\Delta|}, 1 \right\}$$

if $\Delta \neq 0$, and $\bar{\gamma} = 1$ if $\Delta = 0$. Let $x' \in R^{n+1}$ and $x'' \in R^{n+1}$ be defined as

$$x'_i = \begin{cases} y'_i(\bar{\gamma}) \text{ if } & i \neq v, w, \\ 1 \quad \text{if } & i = v, \\ 0 \quad \text{if } & i = w \end{cases} \qquad \text{and} \qquad x''_i = \begin{cases} y''_i(\bar{\gamma}) \text{ if } & i \neq v, w, \\ 1 \quad \text{if } & i = v, \\ 0 \quad \text{if } & i = w. \end{cases}$$

We have that $x = (x' + x'')/2$. In order to verify that x' belongs to $P(T_{n+1})$, we have to verify that the inequalities (1) and (2) are satisfied. The only non trivial check concerns the proof that $1 \leq \sum_{j \in S(u)} x'_j \leq d(u)$, which is true because

$$\begin{aligned}\sum_{j \in S(u)} x'_j &= \sum_{j \in S(u) \setminus \{v\}} x'_j + x'_v = \sum_{j \in S(u) \setminus \{v\}} y'_j(\bar{\gamma}) + 1 = \sum_{j \in S(u) \setminus \{v\}} (y_j + \bar{\gamma}\, \delta_j) + 1 \\ &= \sum_{j \in S(u) \setminus \{v\}} (x_j + \bar{\gamma}\, \delta_j) + 1 = \sum_{j \in S(u) \setminus \{v\}} x_j + \bar{\gamma}\, \Delta + 1 = \sum_{j \in S(u)} x_j + \bar{\gamma}\, \Delta\end{aligned}$$

and by the definition of $\bar{\gamma}$. Analogously we can verify that $x'' \in P(T_{n+1})$.

By Theorem 5, solving the linear programming model

$$\min \sum_{i \in V} x_i \tag{6}$$

$$\text{s.t. (1) and (2)} \tag{7}$$

provides a method for finding a domatic bipartition of a tree having the minimum or maximum number of black nodes. Notice that the same problem can be solved in linear time using the algorithm proposed in Cockayne and Hedetniemi [1975] to find the dominating set of minimum cardinality in a tree. In fact, it easy to see that the complement of any minimal (in the sense of inclusion) dominating set, is also dominating Ore [1962], so that finding a minimum cardinality dominating set ends up with a domatic bipartition where the set of black (resp. white) nodes has minimum (resp. maximum) cardinality.

The integrality property of $P(T)$ does not hold for the polytope

$$P_p(T) = \left\{ x \in P(T) : \sum_{i \in V} x_i = p \right\},$$

obtained by intersecting $P(T)$ with the cardinality constraint (4).

For instance, given the tree $\overline{T} = (\overline{V}, \overline{E})$, with $\overline{V} = \{1, \ldots, 6\}$ and $\overline{E} = \{(1,2), (1,3), (1,4), (4,5), (5,6)\}$, the polyhedron $P(\overline{T})$ has dimension 3 and can be depicted as in Figure 1(a). It has six binary vertices, and the tree $\overline{T}$ has domatic bipartitions with $p = 2, 3, 4$ black nodes. The cardinality constraint $\sum_{i \in V} x_i = 3$ cuts the polyhedron $P(\overline{T})$ as shown in Figure 1(b), introducing two fractional vertices in $P_3(\overline{T})$, $(1/2, 1/2, 1/2, 1/2, 1, 0)$ and $(1/2, 1/2, 1/2, 1/2, 0, 1)$.

As a consequence, the proposed model (1)...(4) for CDBP cannot be solved by linear programming, not even for trees.

5 A polynomial algorithm on trees

In this section we propose a polynomial algorithm for CDBP on trees.

Given a tree $T = (V, E)$, the proposed algorithm arbitrarily selects a node v_0 and considers the tree as rooted in v_0. Initially no node is colored. The algorithm simultaneously solves CDBP for all integer values of p using a backward dynamic programming approach. It examines, one by one, all nodes, starting from the leaves and then considering any node whose children have already been examined. The algorithm examines v_0 as the last node and then outputs the set of values of p for which CDBP has a solution, and for each value p in this set it provides a domatic bipartition of T with p black nodes.

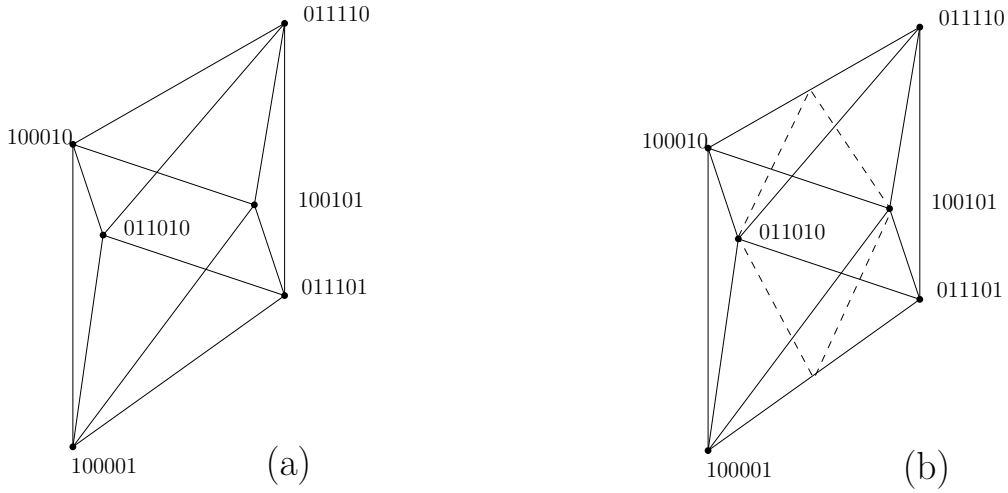

Fig. 1. A representation of $P(\overline{T})$ (a), and its intersection with a cardinality constraint (b).

Let us first give the notation and a rather informal and rough illustration of the key points of the algorithm before providing a formal and rigorous description.

For any $v \in V$ let T_v be the subtree of T containing v and all its descendants. Obviously, $T_{v_0} = T$ and, if $v \neq v_0$ is a leaf, then $T_v = (\{v\}, \emptyset)$. For all $v \neq v_0$, let $f(v)$ be the parent of v and, for any node v that is not a leaf, let $s_1(v), s_2(v), \ldots, s_{q(v)}(v)$ be the children of v, arbitrarily ordered. To avoid cumbersome notation, we will write f, s_j and q instead of $f(v)$, $s_j(v)$ and $q(v)$. The context will clarify the actual meaning. In the sequel we will use several times the set of integers $\{0, 1, \ldots, \lfloor n/2 \rfloor\}$ denoted, for ease of notation, by $\bar{N}$.

Without loss of generality, we may assume that the smallest dominating set is colored black, and hence $1 \leq p \leq \lfloor n/2 \rfloor$.

For any node v denote by $\psi(v)$ the color (black or white) assigned to node v. At the beginning of the algorithm, all nodes are not yet colored.

We give the following definitions.

- Given a coloring ψ of T, we say that a node v is *covered* if at least one of its adjacent nodes has a color different from $\psi(v)$.
- For any node v let $T(v, \ell)$ be the maximal subtree of T_v containing the first ℓ children of v ($1 \leq \ell \leq q$) but none of the children $s_{\ell+1}, s_{\ell+2}, \ldots, s_q$. Evidently $T(v, q) = T_v$.
- For any set S of nodes, denote by $\psi(S)$ the map that assigns to every node v in S a color $\psi(v)$. A coloring $\psi(S)$ is a *k-feasible coloring* for a set of nodes S if all nodes in S are colored so that the number of black nodes in S is k and each node in S is adjacent to at least one node of different color in S.
  We extend this definition by saying that a k-feasible coloring for a set of nodes S is a k-feasible coloring for the subgraph G_S induced by S in tree T.
  More generally, given two sets of colored nodes S and $\tilde{S}$, we say that the coloring $\psi(S)$ is *k-feasible for S (or G_S) given $\psi(\tilde{S})$* if the number of black nodes in S is k and each node in S is adjacent to at least one node in $S \cup \tilde{S}$ of different color.
  For instance, if $v \neq v_0$ is not a leaf and $\tilde{S} = \{v, f\}$ with $\psi(v) = \psi(f) = w$, then it is necessary that at least one of the children of v be colored black in order to get a k-feasible coloring of T_v given $\psi(\tilde{S})$,

for some k. However, if the color of f changes to black, then there could be a k-feasible coloring of T_v given $\psi(\tilde{S})$ having all children of v colored white. In the tree $T = (V, E)$ of Figure 2, the given coloring is 3-feasible. The coloring of subtree T_2 is not 2-feasible, because node 2 is not covered in T_2. However, it is 2-feasible for $V(T_2)$ given $\psi(1)$ and $\psi(2)$.

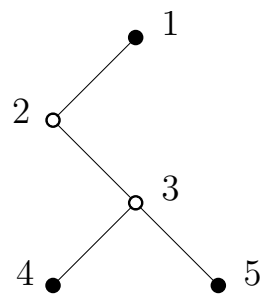

Fig. 2. A sample tree.

- For any node v let $K(T(v, \ell) \mid \psi(v))$ be the set of values $k \in \bar{N}$ such that there exists a k-feasible coloring for $T(v, \ell)$ given $\psi(v)$; note that in any such k-feasible coloring, node v is covered by at least one of its first ℓ children.
- For any $k \in K(T(v, \ell) \mid \psi(v))$, let $S(T(v, \ell), k \mid \psi(v))$ be any of the sets of k black nodes in a k-feasible coloring for $T(v, \ell)$ given $\psi(v)$.

The algorithm provides the set P of values p for which CDBP has a solution. Since $T = T(v_0, q)$, the algorithm, at the end, computes the elements of the sets $K(T(v_0, q) \mid \psi(v_0) = b)$ and $K(T(v_0, q) \mid \psi(v_0) = w)$ whose union gives P. Moreover, for any $p \in P$ the algorithm also finds a partition of V into two dominating sets V' and V'', such that $|V'| = p$. Indeed, if $p \in K(T(v_0, q) \mid \psi(v_0) = b)$ then V' can be chosen as $S(T(v_0, q), p \mid \psi(v_0) = b)$, and if $p \in K(T(v_0, q) \mid \psi(v_0) = w)$ then V' can be chosen as $S(T(v_0, q), p \mid \psi(v_0) = w)$.

To compute the sets $K(T(v_0, q) \mid \psi(v_0))$ we consider all the children $s_1, \ldots, s_q$ of v_0 and the related trees $T_{s_1}, \ldots, T_{s_q}$.

Take for instance the case $\psi(v_0) = b$. Since v_0 is colored black, at least one of its children must be white, so every coloring for $s_1, \ldots, s_q$ is acceptable, except the case all black. Any constrained domatic bipartition of the tree T induces a k_j-feasible coloring of tree T_{s_j} given $\psi(s_j)$ and $\psi(v_0)$ for some k_j and for any $j = 1, \ldots, q$. However, any black s_j must be covered by one of its children, while any white s_j is covered by its parent v_0, and so it can have arbitrarily colored children. It follows that, to recursively compute the elements of set $K(T(v_0, q) \mid \psi(v_0) = b)$, we need the sets $K(T_{s_j} \mid \psi(s_j) = b)$, but also another type of set, see K^+ defined below, that depends on the colors of both s_j and its parent.

- For any node $v \neq v_0$ let $K^+(T_v \mid \psi(v), \psi(f))$ be the set of values $k \in \bar{N}$ such that there exists a k-feasible coloring for T_v, given $\psi(v)$ and $\psi(f)$.
  Notice that $K(T_v \mid \psi(v)) \subseteq K^+(T_v \mid \psi(v), \psi(f))$ for all $\psi(f)$.
- For any $k \in K^+(T_v \mid \psi(v), \psi(f))$, let $S^+(T_v, k \mid \psi(v), \psi(f))$ be any of the sets of k black nodes in a k-feasible coloring for T_v, given $\psi(v)$ and $\psi(f)$.
  Notice that if $k \notin K^+(T_v \mid \psi(v), \psi(f))$, then $S^+(T_v, k \mid \psi(v), \psi(f))$ is not defined, and, moreover, if 0 belongs to $K^+(T_v \mid \psi(v), \psi(f))$, then $S^+(T_v, 0 \mid \psi(v), \psi(f)) = \emptyset$.

Notice that K^+ and K could be merged into a single definition, the difference being only on the nodes whose color is conditioning, only node v for set K, and nodes v and f for set K^+. However, we prefer to use two separate notations to stress the fact that in K^+ we condition upon the color of f, parent of v.

Using these latter definitions, $K(T(v_0, q) \mid \psi(v_0) = b)$ can be deduced from $K^+(T_{s_j} \mid \psi(s_j) = b, \psi(v_0) = b)$ and $K^+(T_{s_j} \mid \psi(s_j) = w, \psi(v_0) = b)$, for $j = 1, \ldots, q$.

The elements of the set $K^+(T_{s_j} \mid \psi(s_j) = b, \psi(v_0) = b)$ are recursively computed through the analysis of each child of s_j, noting that at least one of the children of s_j must be colored white, since both s_j and its parent v_0 are black.

The elements of the set $K^+(T_{s_j} \mid \psi(s_j) = w, \psi(v_0) = b)$ are recursively computed through the analysis of each child of s_j, noting that s_j is covered by its parent v_0. Therefore, its children can be arbitrarily colored.

Once s_j is a leaf, we can directly compute the sets $K^+(T_{s_j} \mid \psi(s_j), \psi(f))$ for all the combinations of colors.

Since for any node $v \in V$ the sets $K(T_v \mid \psi(v))$ are computed through $K(T(v, \ell) \mid \psi(v))$ setting ℓ equal to 1 up to the number of children of v, we also need the following sets K^- and S^-.

- For any node v let $K^-(T(v, \ell) \mid \psi(v))$ be the set of values $k \in \bar{N}$ such that (i) all children $s_1, s_2, \ldots, s_\ell$ have the same color as v; (ii) there exist a k-feasible coloring for $T(v, \ell) \setminus \{v\}$ if $\psi(v) = w$ and a $(k-1)$-feasible coloring for $T(v, \ell) \setminus \{v\}$ if $\psi(v) = b$.
- For any $k \in K^-(T(v, \ell) \mid \psi(v))$, let $S^-(T(v, \ell), k \mid \psi(v))$ be any of the sets of k black nodes in a coloring of $T(v, \ell)$ that satisfies (i) and (ii) in the previous definition.
  For instance, in the graph of Figure 2 with no colors on the nodes, the set $K^-(T(3, 2) \mid \psi(3))$ is empty, since nodes 3, 4 and 5 should have the same color, and condition (ii) could not be satisfied. Furthermore, $K^-(T(2, 1) \mid \psi(2) = b) = \{2\}$ and $S^-(T(2, 1), k = 2 \mid \psi(2) = b) = \{2, 3\}$.

The following operation is needed to compute the elements of the set $K(T(v, \ell) \mid \psi(v))$ by combining the information contained in sets $K(T(v, \ell - 1) \mid \psi(v))$, $K^-(T(v, \ell - 1) \mid \psi(v))$ and $K^+(T_{s_\ell} \mid \psi(s_\ell), \psi(v))$.

- For any $A, B \subseteq \bar{N}$ let $A \oplus B = \{a + b \mid a \in A, b \in B, a + b \leq n/2\} \subseteq \bar{N}$. Notice that if A or B is empty, so is $A \oplus B$. Notice also that $A \oplus B$ could be empty even if both A and B are non empty: consider, for instance, the case where $n = 10$ and $A = B = \{3, 4, 5\}$.

The algorithm explores the rooted tree T analyzing the nodes from bottom to top, starting from the leaves. In every iteration it examines a node v such that all its children have already been examined, and it computes the elements of the sets $K^+(T_v \mid \psi(v), \psi(f))$ for any possible choice of the colors of node v and its parent f.

Let us now provide a formal description of the algorithm.

Initialization

Let v_0 be an arbitrary node and consider T as rooted in v_0. For any node that is not a leaf, order arbitrarily all its children.

Declare all nodes as not examined.

For any $v \in V \setminus \{v_0\}$ that is a leaf, consider $T_v = (\{v\}, \emptyset)$. We compute the sets K^+ and, when defined, S^+ for the following four cases depending on the colors of v and $f = f(v)$:

- $\psi(v) = b$ and $\psi(f) = b$. Then: $K^+(T_v \mid \psi(v), \psi(f)) = \emptyset$;
- $\psi(v) = b$ and $\psi(f) = w$. Then:
  $K^+(T_v \mid \psi(v), \psi(f)) = \{1\}$ and $S^+(T_v, k = 1 \mid \psi(v), \psi(f)) = \{v\}$;

- $\psi(v) = w$ and $\psi(f) = b$. Then:
  $K^+(T_v \mid \psi(v), \psi(f)) = \{0\}$ and $S^+(T_v, k = 0 \mid \psi(v), \psi(f)) = \emptyset$;
- $\psi(v) = w$ and $\psi(f) = w$. Then: $K^+(T_v \mid \psi(v), \psi(f)) = \emptyset$.

Declare v examined.

Main iteration

Step 1
Select a node v not yet examined, such that all of its q children have been examined.
Let $\ell = 1$.

Case 1.1: $\psi(v) = b$.
Let $K(T(v,1) \mid \psi(v) = b) = \{1\} \oplus K^+(T_{s_1} \mid \psi(s_1) = w, \psi(v) = b)$.
If this set is nonempty, for each $k \in K(T(v,1) \mid \psi(v) = b)$,
let $S(T(v,1), k \mid \psi(v) = b)$
$= \{v\} \cup S^+(T_{s_1}, k-1 \mid \psi(s_1) = w, \psi(v) = b)$.
Let $K^-(T(v,1) \mid \psi(v) = b) = \{1\} \oplus K^+(T_{s_1} \mid \psi(s_1) = b, \psi(v) = b)$.
If this set is nonempty, for each $k \in K^-(T(v,1) \mid \psi(v) = b)$,
let $S^-(T(v,1), k \mid \psi(v) = b)$
$= \{v\} \cup S^+(T_{s_1}, k-1 \mid \psi(s_1) = b, \psi(v) = b)$.

Case 1.2: $\psi(v) = w$.
Let $K(T(v,1) \mid \psi(v) = w) = K^+(T_{s_1} \mid \psi(s_1) = b, \psi(v) = w)$.
If this set is nonempty, for each $k \in K(T(v,1) \mid \psi(v) = w)$,
let $S(T(v,1), k \mid \psi(v) = w)$
$= S^+(T_{s_1}, k \mid \psi(s_1) = b, \psi(v) = w)$.
Let $K^-(T(v,1) \mid \psi(v) = w) = K^+(T_{s_1} \mid \psi(s_1) = w, \psi(v) = w)$.
If this set is nonempty, for each $k \in K^-(T(v,1) \mid \psi(v) = w)$,
let $S^-(T(v,1), k \mid \psi(v) = w)$
$= S^+(T_{s_1}, k \mid \psi(s_1) = w, \psi(v) = w)$.

If $\ell = q$ and $v \neq v_0$, go to Step 3.
If $\ell = q$ and $v = v_0$, go to Step 4.

Step 2
Let $\ell = \ell + 1$.

Case 2.1: $\psi(v) = b$.

Let $K(T(v,\ell) \mid \psi(v) = b)$
$= K(T(v,\ell-1) \mid \psi(v) = b) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell) = w, \psi(v) = b)$
$\cup K(T(v,\ell-1) \mid \psi(v) = b) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell) = b, \psi(v) = b)$
$\cup K^-(T(v,\ell-1) \mid \psi(v) = b) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell) = w, \psi(v) = b)$.
If this set is nonempty, for each $k \in K(T(v,\ell) \mid \psi(v) = b)$, it must be $k = k' + k''$ with
2.1.a) $k' \in K(T(v,\ell-1) \mid \psi(v) = b)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell) = w, \psi(v) = b)$ or
2.1.b) $k' \in K(T(v,\ell-1) \mid \psi(v) = b)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell) = b, \psi(v) = b)$ or
2.1.c) $k' \in K^-(T(v,\ell-1) \mid \psi(v) = b)$ but $k' \notin K(T(v,\ell-1) \mid \psi(v) = b)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell) = w, \psi(v) = b)$.

In case 2.1.a, let $S(T(v,\ell),k \mid \psi(v)=b)$
$= S(T(v,\ell-1),k' \mid \psi(v)=b) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=w,\psi(v)=b)$.
In case 2.1.b, let $S(T(v,\ell),k \mid \psi(v)=b)$
$= S(T(v,\ell-1),k' \mid \psi(v)=b) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=b,\psi(v)=b)$.
In case 2.1.c, let $S(T(v,\ell),k \mid \psi(v)=b)$
$= S^-(T(v,\ell-1),k' \mid \psi(v)=b) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=w,\psi(v)=b)$.

Let $K^-(T(v,\ell) \mid \psi(v)=b)$
$= K^-(T(v,\ell-1) \mid \psi(v)=b) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell)=b,\psi(v)=b)$.
If this set is nonempty, for each $k \in K^-(T(v,\ell) \mid \psi(v)=b)$, it must be $k=k'+k''$ with $k' \in K^-(T(v,\ell-1) \mid \psi(v)=b)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell)=b,\psi(v)=b)$.
Let $S^-(T(v,\ell),k \mid \psi(v)=b)$
$= S^-(T(v,\ell-1),k' \mid \psi(v)=b) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=b,\psi(v)=b)$.

Case 2.2: $\psi(v)=w$.

Let $K(T(v,\ell) \mid \psi(v)=w)$
$= K(T(v,\ell-1) \mid \psi(v)=w) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell)=b,\psi(v)=w)$
$\cup K(T(v,\ell-1) \mid \psi(v)=w) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell)=w,\psi(v)=w)$
$\cup K^-(T(v,\ell-1) \mid \psi(v)=w) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell)=b,\psi(v)=w)$.
If this set is nonempty, for each $k \in K(T(v,\ell) \mid \psi(v)=w)$, it must be $k=k'+k''$ with
2.2.a) $k' \in K(T(v,\ell-1) \mid \psi(v)=w)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell)=b,\psi(v)=w)$ or
2.2.b) $k' \in K(T(v,\ell-1) \mid \psi(v)=w)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell)=w,\psi(v)=w)$ or
2.2.c) $k' \in K^-(T(v,\ell-1) \mid \psi(v)=w)$ but $k' \notin K(T(v,\ell-1) \mid \psi(v)=w)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell)=b,\psi(v)=w)$.
In case 2.2.a, let $S(T(v,\ell),k \mid \psi(v)=w)$
$= S(T(v,\ell-1),k' \mid \psi(v)=w) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=b,\psi(v)=w)$.
In case 2.2.b, let $S(T(v,\ell),k \mid \psi(v)=w)$
$= S(T(v,\ell-1),k' \mid \psi(v)=w) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=w,\psi(v)=w)$.
In case 2.2.c, let $S(T(v,\ell),k \mid \psi(v)=w)$
$= S^-(T(v,\ell-1),k' \mid \psi(v)=w) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=b,\psi(v)=w)$.

Let $K^-(T(v,\ell) \mid \psi(v)=w)$
$= K^-(T(v,\ell-1) \mid \psi(v)=w) \oplus K^+(T_{s_\ell} \mid \psi(s_\ell)=w,\psi(v)=w)$.
If this set is nonempty, for each $k \in K^-(T(v,\ell) \mid \psi(v)=w)$, it must be $k=k'+k''$ with $k' \in K^-(T(v,\ell-1) \mid \psi(v)=w)$ and $k'' \in K^+(T_{s_\ell} \mid \psi(s_\ell)=w,\psi(v)=w)$.
Let $S^-(T(v,\ell),k \mid \psi(v)=w)$
$= S^-(T(v,\ell-1),k' \mid \psi(v)=w) \cup S^+(T_{s_\ell},k'' \mid \psi(s_\ell)=w,\psi(v)=w)$.

If $\ell < q$, go to Step 2.
If $\ell = q$ and $v = v_0$ go to Step 4.

Step 3

Case 3.1: $\psi(v)=b$ and $\psi(f)=b$.
Set $K^+(T_v \mid \psi(v)=b,\psi(f)=b) = K(T_v \mid \psi(v)=b)$
and, for any $k \in K^+(T_v \mid \psi(v)=b,\psi(f)=b)$ set
$S^+(T_v,k \mid \psi(v)=b,\psi(f)=b) = S(T_v,k \mid \psi(v)=b)$.

Case 3.2: $\psi(v) = b$ and $\psi(f) = w$.
Set $K^+(T_v \mid \psi(v) = b, \psi(f) = w)$
$= K(T_v \mid \psi(v) = b) \cup K^-(T_v \mid \psi(v) = b)$.
For any $k \in K^+(T_v \mid \psi(v) = b, \psi(f) = w)$, if $k \in K(T_v \mid \psi(v) = b)$ set $S^+(T_v, k \mid \psi(v) = b, \psi(f) = w) = S(T_v, k \mid \psi(v) = b)$,
else set $S^+(T_v, k \mid \psi(v) = b, \psi(f) = w) = S^-(T_v, k \mid \psi(v) = b)$.

Case 3.3: $\psi(v) = w$ and $\psi(f) = b$.
Set $K^+(T_v \mid \psi(v) = w, \psi(f) = b)$
$= K(T_v \mid \psi(v) = w) \cup K^-(T_v \mid \psi(v) = w)$.
For any $k \in K^+(T_v \mid \psi(v) = w, \psi(f) = b)$, if $k \in K(T_v \mid \psi(v) = w)$ set $S^+(T_v, k \mid \psi(v) = w, \psi(f) = b) = S(T_v, k \mid \psi(v) = w)$,
else set $S^+(T_v, k \mid \psi(v) = w, \psi(f) = b) = S^-(T_v, k \mid \psi(v) = w)$.

Case 3.4: $\psi(v) = w$ and $\psi(f) = w$.
Set $K^+(T_v \mid \psi(v) = w, \psi(f) = w)$
$= K(T_v \mid \psi(v) = w)$
and, for any $k \in K^+(T_v \mid \psi(v) = w, \psi(f) = w)$ set
$S^+(T_v, k \mid \psi(v) = w, \psi(f) = w) = S(T_v, k \mid \psi(v) = w)$.

Declare v examined.
Go to Step 1.

(End of the Main Iteration)

Step 4 (Final Step)

Case 4.1: $\psi(v_0) = b$.
Set $K_b = K(T(v_0, q) \mid \psi(v_0) = b)$
and, for any $k \in K_b$, set $S_b(k) = S(T(v_0, q), k \mid \psi(v_0) = b)$.

Case 4.2: $\psi(v_0) = w$.
Set $K_w = K(T(v_0, q) \mid \psi(v_0) = w)$
and, for any $k \in K_w$, set $S_w(k) = S(T(v_0, q), k \mid \psi(v_0) = w)$.

Declare v_0 examined.

The set V can be partitioned into two dominating sets V' and V'', whose cardinality is p and $n - p$ (with $p \leq n - p$), if and only if $p \in P = K_b \cup K_w$. If $p \in K_b$ (resp. $p \in K_w$) then two dominating sets are provided by $S_b(p)$ (resp. $S_w(p)$) and $V \setminus S_b(p)$ (resp. $V \setminus S_w(p)$). An implementation of the algorithm is available at Andreatta et al. [2015].

Applying the algorithm to the tree of Figure 2 with no coloring, the final step gives $K_b = \{2\}$ and $S_b(2) = \{1, 3\}$, $K_w = \{2\}$ and $S_w(2) = \{2, 3\}$. The detailed trace of the execution is available at Andreatta et al. [2015].

This means that CDBP has a solution for p equal to 2 and the tree has (at least) two domatic bipartitions, one with $V' = \{1, 3\}$ and $V'' = \{2, 4, 5\}$, the other with $V' = \{2, 3\}$ and $V'' = \{1, 4, 5\}$.

Notice that other domatic bipartitions of the tree can be obtained from the previous ones, by switching colors black and white. Therefore, the given tree also admits domatic bipartitions with 3 black nodes.

Proposition 6 *The overall worst case complexity of the algorithm is $O(n^3)$.*

Proof. The algorithm starts from the leaves of a rooted tree and visits one by one the nodes, once all children of a node have been visited. When a node is visited, all of its descendant edges are processed. Hence globally, all the $n-1$ edges of the tree are processed once and only once. Each edge processing essentially requires computing sums $A \oplus B$ and set unions. If we implement all sets as boolean n-vectors the sum $A \oplus B$ costs $O(n^2)$ (the sets are not necessarily made up of contiguous integer numbers). The union of two sets costs $O(n)$ and it has to be computed at most n times (for each value of a feasible k) for a total cost $O(n^2)$. Hence we have an overall worst case complexity $O(n^3)$.
Note that keeping track of the values k' and k'' such that $k = k' + k''$ (in cases 2.1 and 2.2) can be done at the very moment when the sums $A \oplus B$ are computed. Hence when a feasible k is found one should also store the two values k' and k'' together with the condition that has allowed for such a sum.

6 Weighted CDBP

A possible extension of CDBP is the *Weighted Constrained Domatic Bipartition Problem*: we consider a graph $G = (V, E)$ with weight $w(v)$ on each node v and ask if V can be partitioned into two disjoint dominating sets V' and V'' such that $\sum_{v \in V'} w(v)$ is equal to a given value Q and, in case of positive answer, provide such partition. As shown in the following, this problem is NP-hard, even for trees. Clearly, CDBP on trees is a polynomially solvable special case of the weighted version with $w(v) = 1$, for every node v.

Proposition 7 *The Weighted Constrained Domatic Bipartition Problem is NP-hard, even for trees.*

Proof. We prove the assertion via a transformation from the notoriously NP-hard Number Partition Problem [Karp, 1972]. Given n positive integers $a_1, a_2, \ldots, a_n$, we recall that the Number Partition (or, simply, Partition) Problem asks to determine a partition, if it exists, of $\{1, 2, \ldots, n\}$ into two subsets J_1 and J_2 such that $\sum_{j \in J_1} a_j = \sum_{j \in J_2} a_j$ (assuming that the sum of all numbers is even).
The Partition Problem can be polynomially transformed into the Weighted Constrained Domatic Bipartition Problem on a tree with $Q = n + \frac{1}{2} \sum_{j=1}^{n} a_j$.
The tree is characterized as follows: consider a path containing n nodes $u_1, u_2, \ldots, u_n$, each with weight 1 and to each node u_i attach an edge (u_i, v_i) where v_i has weight $a_i + 1$. It is not difficult to see that the resulting graph is a tree. The tree has the form of a comb and can be partitioned into two dominating sets of equal weight if and only if the original Number Partition instance has a solution.
This proves that the Weighted Constrained Domatic Bipartition Problem is NP-hard.

Notice that, from Theorem 5, finding a domatic bipartition having one set of minimum weight is polynomial on trees.

Assuming integer nonnegative weights for nodes, a pseudo-polynomial algorithm for Weighted CDBP on trees can be obtained from the one proposed in Section 5 for the non-weighted case, as follows.

First, we adapt the notion of feasible coloring. For any set S of colored nodes, we say that a coloring $\psi(S)$ is a *k-weighted feasible coloring* for S if the total weight of the black nodes in S is k and each node in S is adjacent to at least one node of different color in S. More generally, given two sets of colored nodes S and $\tilde{S}$, we say that the coloring $\psi(S)$ is a *k-weighted feasible coloring for S given $\psi(\tilde{S})$* if the total weight of the black nodes in S is k and each node in S is adjacent to at least one node in $S \cup \tilde{S}$ of different color.

Set $W = \left\lfloor \frac{1}{2} \sum_{v \in V} w(v) \right\rfloor$ and let $\bar{W} = \{0, 1, \ldots, W\}$ be the set of all possible total weights for the black nodes. We remark that, similarly to the non-weighted case, we can restrict ourselves to consider the total weight of the black nodes lower than the total weight of the white nodes.

For any node v, redefine $K(T(v, \ell) \mid \psi(v))$ as the set of values $k \in \bar{W}$ such that there exists a k-weighted feasible coloring for $T(v, \ell)$ given $\psi(v)$. Similar to K, we modify the definitions of the sets K^+, K^-, and the related S, S^+ and S^-. In this way, we store information on the possible total weights of the black nodes in feasible colorings (and the colorings themselves), instead of information on the cardinality of the same sets.

For any $A, B \subseteq \bar{W}$ the operation $A \oplus B$ has to be redefined equal to $\{a+b \mid a \in A, b \in B, a+b \leq W\} \subseteq \bar{W}$.

Finally, we modify the algorithm for the non-weighted case by replacing, where needed, the value '1' with $w(v)$: for example, in the initialization step, the value $w(v)$ (instead of 1) is stored in $K^+(T_v \mid \psi(v), \psi(f))$ in case $\psi(v) = b$ and $\psi(f) = w$.

The resulting algorithm simultaneously solves the Weighted CDBP on trees for all the weights between 0 and W in $O(nW^2 + n^2W)$: in fact, for each of the $n-1$ edges, the algorithm computes a constant number of sums $A \oplus B$, with $A, B \subseteq \bar{W}$ (cost $O(W^2)$), and performs at most W times the union of two subsets of V (cost $O(nW)$).

7 Conclusions

We consider the Constrained Domatic Bipartition Problem (CDBP), i.e., the problem of determining a bipartition, if it exists, of the nodes of a graph into two dominating sets, with the additional constraint that one of the two subsets has a given cardinality. The problem is NP-hard and we have formulated it as an integer linear program. In this paper we have focused on trees showing that, in this case, the polyhedron associated to the formulation without the cardinality constraint has integral vertices. This property is lost when adding the cardinality constraint. This is the motivation for providing a polynomial dynamic programming algorithm for CDBP on trees, whose computational complexity is $O(n^3)$, n being the number of nodes. We have also considered the weighted version of CDBP, showing it is NP-Hard even on trees, and discussing on how the previous algorithm can be adapted to form a pseudo-polynomial algorithm for this more general problem.

Future work will concentrate on computational approaches for solving the problem in any graph and on exploring other problems that are NP-hard on graphs but polynomially solvable on trees.

CDBP can be reformulated as the problem of finding a star covering with a prescribed number of stars. We are currently investigating the problem of finding a partition (rather than a covering) of a graph into stars, satisfying given cardinality constraints. This problem is intimately connected to CDBP and, yet, the results from CDBP cannot be automatically transferred to the star partition problem.

Acknowledgments

We thank Michele Conforti for pointing out the transformation from Number Partition to Weighted Constrained Domatic Bipartition problems.

We also thank two anonymous Referees for their detailed suggestions that greatly improved the readability of the paper.

Bibliography

O. Ore, *Theory of Graphs*, vol. XXXVIII, American Mathematical Society, 1962.

E. J. Cockayne, S. T. Hedetniemi, Towards a Theory of Domination in Graphs, *Networks* 7 (1977) 247–261.

G. Andreatta, L. De Giovanni, P. Serafini, Optimal Shift Partitioning of Pharmacies, *Computers & Operations Research* 55 (2015) 88–98.

G. Andreatta, C. De Francesco, L. De Giovanni, P. Serafini, The Constrained Star Partition Problem, *Unpublished results*, 2016.

E. J. Cockayne, S. T. Hedetniemi, A Linear Algorithm for the Domination Number of a Tree, *Information Processing Letters* 4 (1975) 41–44.

M. R. Garey, D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness*, W. H. Freeman and Co., San Francisco, 1979.

G. K. Manacher, T. A. Mankus, Finding a Domatic Partition of an Interval Graph in time O(n), *SIAM Journal on Discrete Mathematics* 9 (1996) 167–172.

S. Poon, W. C. Yen, C. Ung, Domatic Partition on Several Classes of Graphs, *Lecture Notes in Computer Science* 7402 (2012) 245–256.

G. Andreatta, L. De Giovanni, P. Serafini, Optimal Shift Coloring of Trees, *Operations Research Letters* 42 (2014) 251–256, doi:doi:10.1016/j.orl.2014.04.004.

G. J. Chang, The Domatic Number Problem, *Discrete Mathematics* 125 (1994) 115–122.

G. Andreatta, C. De Francesco, L. De Giovanni, F. Rampado, P. Serafini, http://www.math.unipd.it/∼luigi/manuscripts/CDBP-tree-algorithm, 2015.

R. M. Karp, Reducibility among Combinatorial Problems, in: R. E. Miller, J. W. Tatcher (Eds.), *Complexity of Computer Computations*, Plenum Press, New York, 85–103, 1972.