Solving some NP-complete problems using split decomposition [1]

Michaël Rao

CNRS - LIRMM, Université Montpellier II,
161 rue Ada,
34392 Montpellier Cedex 5 - France

Abstract

We show how to use the split decomposition to solve some NP-hard optimization problems on graphs. We give algorithms for clique problem and domination-type problems. Our main result is an algorithm to compute a coloration of a graph using its split decomposition. Finally we show that the clique-width of a graph is bounded if and only if the clique-width of each representative graph in its split decomposition is bounded.

Key words: Graph algorithms, graph coloring, NP-complete problems, split decomposition, clique-width.

1 Introduction

Decompositions play an important role in graph theory. The central role of decompositions in the recent proof of Claude Berge's Strong Perfect Graph Conjecture is an exciting example [7]. Furthermore various decompositions of graphs such as decomposition by clique separators, tree decomposition and clique decomposition are often used to design efficient graph algorithms. There are beautiful general results stating that a variety of NP-complete graph problems can be solved in linear time for graphs of bounded treewidth and bounded clique-width [1,10].

Email address: `Michael.Rao@lirmm.fr` (Michaël Rao).

[1] Extended version of *Coloring a Graph Using Split Decomposition*, in WG'04, Lecture Notes in Computer Science 3353, Springer-Verlag, Berlin, 2004, pp.129–141.

The typical approach to design efficient algorithms using graph decompositions works as follows. The algorithm recursively decomposes the graph into smaller graphs, until the obtained graphs cannot be decomposed further. Such graphs are called prime. Then the algorithm solves the problem on the prime graphs, and recursively combines the solutions to find the solution for the original graph. In order to obtain an efficient algorithm with this approach, the input graph has to be restricted to a graph class nicely decomposable with respect to the decomposition.

Several decompositions have been studied in this direction. Tarjan showed that various NP-complete problems can be solved using decomposition by clique separators [24]. Modular decomposition for discrete structures is known and studied for a long time. A nice survey on this topic has been written by Möhring and Radermacher [22]. Recently, a lot of work has been done studying modular decomposition on graphs. This includes linear time modular decomposition algorithms [12,21,25], the study of modular decomposition for certain graph classes [6], and efficient algorithms for some NP-complete graph problems using modular decomposition [4,3,5]. Finally many NP-complete graph problems can be solved by polynomial or linear time algorithms using tree decomposition or clique decomposition, if the treewidth or the clique-width of the graph is bounded, and the graph is given with a clique decomposition in the case of the clique-width [1,10,15,19].

In this paper, we consider the split decomposition (sometimes also called join decomposition) which can be seen as a generalization of the modular decomposition. There are only few papers presenting algorithms for NP-complete problems using split decomposition. Cunningham gave an algorithm for the independent set problem [13]. Cicerone and Di Stefano showed how to apply this algorithm to parity graphs, which are exactly those graphs for which all representative graphs with respect to the split decomposition are bipartite or complete [8]. They presented also an algorithm for the clique problem, but unfortunately this algorithm is flawed. Split decomposition was also used for circle graph recognition [16,23] and parity graph recognition [8,14].

This paper is organized as follows: Section 2 gives several preliminaries and Section 3 presents the split decomposition. In Section 4 we recall Cunningham's algorithm for the independent set problem and we give a correct algorithm for the clique problem. Section 5 is dedicated to domination problems. In Section 6 we present an algorithm to compute the chromatic number using split decomposition. In Section 7 we discuss on the relation between the clique-width of a graph and the clique-width of the prime graphs in its split decomposition. Finally, in Section 8 we apply the previous results to obtain polynomial time algorithms for two nicely decomposable graph classes.

2 Preliminaries

Every graph is supposed to be finite, undirected, simple and loopless. A graph is a pair (V, E) where V is the set of *vertices* (or *nodes*) and $E \subseteq \{\{u, v\} : (u, v) \in V^2 \text{ and } u \neq v\}$ the set of *edges*. If G is a graph, we denote by $V(G)$ its set of vertices and by $E(G)$ its set of edges. Furthermore we denote by $n(G) = |V(G)|$ and $m(G) = |E(G)|$. When G is clear in the context, we simply write V, E, n and m instead of $V(G)$, $E(G)$, $n(G)$ and $m(G)$.

The *neighborhood* of a vertex v in G is $N_G(v) = \{u \in V : \{u, v\} \in E\}$ and its *closed neighborhood* is $N_G[v] = N_G(v) \cup \{v\}$. We shall write $N(v)$ and $N[v]$ if there is no ambiguity. Two vertices u and v are *adjacent* if $\{u, v\} \in E$. A *module* is a set $M \subseteq V$ such that for all $v \in V \setminus M$ either $N(v) \cap M = \emptyset$ or $M \subseteq N(v)$.

Let $G = (V, E)$ and $W \subseteq V$. We denote by $G[W] = (W, \{\{u, v\} \in E : (u, v) \in W^2\})$ the subgraph of G *induced* by W. We denote by $G - W = G[V \setminus W]$ and for $v \in V$ we write $G - v$ instead of $G - \{v\}$. The *complement* of G is the graph $\overline{G} = (V, \overline{E})$ where $\overline{E} = \{\{u, v\} : (u, v) \in V^2, u \neq v \text{ and } \{u, v\} \notin E\}$.

A graph is *complete* if $E = \{\{u, v\} : (u, v) \in V^2 \text{ and } u \neq v\}$. We denote by K_n the complete graph with n vertices. A graph is a *star* if there is a $u \in V$ such that $E = \{\{u, v\} : v \in V \setminus \{u\}\}$. A graph is graph is a *cycle* if there is an ordering $(v_1, \ldots, v_n)$ of vertices of G such that $E = \{v_1, v_n\} \cup \{\{v_i, v_{i+1}\} : i \in \{1, 2 \ldots n - 1\}\}$. We denote by C_n the cycle of n vertices. A *stable set* of G is a subset $S \subseteq V$ such that $G[S]$ has no edges and a *clique* of G is a subset $C \subseteq V$ such that $G[C]$ is complete.

A *weighted graph* is a pair (G, w) where G is a graph, and $w : V \to \mathbb{N}$ is a weight function. The weight of a subset $W \subseteq V$ is $w(W) = \sum_{v \in W} w(v)$. The *weighted stability number* of a weighted graph (G, w) is the maximum weight of a stable set of G, and is denoted by $\alpha_w(G)$. The *weighted clique number*, denoted by $\omega_w(G)$, is the maximum weight of a clique of G.

The *chromatic number* of a graph $G = (V, E)$, denoted by $\chi(G)$, is the smallest integer k such that there is a function $f : V \to \{1, \ldots, k\}$ and for all $u, v \in V$, $\{u, v\} \in E$ we have $f(u) \neq f(v)$. Clearly a *k-coloring* C of G can be seen as a partition of V into k stable sets.

A *multiset* may contain multiple instances of the same element. It is denoted by $\langle e_1, \ldots, e_k \rangle$. A *weighted coloring* of the weighted graph (G, w) (or a *w-weighted coloring* of G) is a multiset C of stable sets of G such that for all $v \in V$, $|\langle S \in C : v \in S \rangle| \geq w(v)$. The *weighted chromatic number* of a weighted graph (G, w), denoted by $\chi_w(G)$, is the minimum cardinality of a w-weighted coloring of G. It is well known that $\chi_w(G) \geq \omega_w(G)$.

Throughout the paper we use the following notations in order to simplify the presentation. For $a, b \in \mathbb{N}$ such that $b \geq a$, $[\![a, b]\!]$ denotes the set $\{a, a+1, \ldots, b\}$. Let $w : V \rightarrow \mathbb{N}$ be a function and let $a \in \mathbb{N}$. We denote $w|_{v \rightarrow a}$ the function on the domain $V \cup \{v\}$ such that $w|_{v \rightarrow a}(v) = a$ and $w|_{v \rightarrow a}(u) = w(u)$ for all $u \in V \setminus \{v\}$. Note that v can be a member of V or not. The composition of functions is denoted by $\circ$.

3 Reviewing split decomposition

A *split* of a connected graph $G = (V, E)$ is a partition of V into two sets V_1 and V_2 such that $|V_1| \geq 2$, $|V_2| \geq 2$ and every vertex in V_1 with a neighbor in V_2 has the same neighborhood in V_2. In others words, $\{V_1, V_2\}$ is a split if there is a $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ such that for all $v \in W_1$, $N(v) \cap V_2 = W_2$ and for all $v \in V_1 \setminus W_1$, $N(v) \cap V_2 = \emptyset$. Following this definition, we define the *simple decomposition* of G by $\{V_1, V_2\}$: G is decomposed into G_1 and G_2, where G_i is the subgraph of G induced by V_i with an additional vertex v such that $N_{G_i}(v) = W_i$ (for $i \in \{1, 2\}$). The vertex v plays a special role and is called a *marker*. Note that G_1 and G_2 are connected. A graph is *prime* if it does not have a split. We say that G is *decomposable* into G_1 and G_2 if there is a split $\{V_1, V_2\}$ such that the simple decomposition of G by $\{V_1, V_2\}$ gives G_1 and G_2.

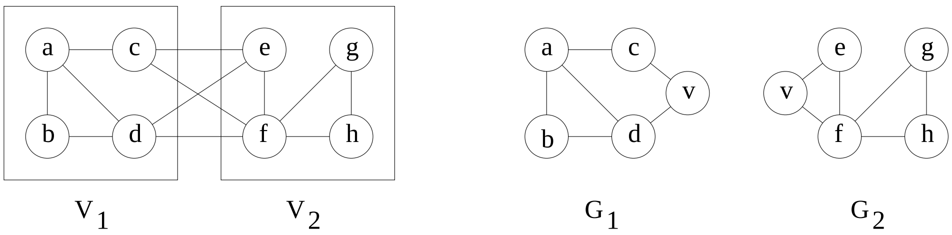

Fig. 1. A graph with a split $\{V_1, V_2\}$ and the simple decomposition by $\{V_1, V_2\}$.

Additionally we define a related composition. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two connected graphs, such that $|V_1 \cap V_2| = 1$. Let v such that $\{v\} = V_1 \cap V_2$. Then $G_1 * G_2$ is the graph with vertex set $(V_1 \cup V_2) \setminus \{v\}$, and edge set $E(G_1 - v) \cup E(G_2 - v) \cup \{\{x, y\} : x \in N_{G_1}(v) \text{ and } y \in N_{G_2}(v)\}$. Obviously if G is decomposable into G_1 and G_2 then $G = G_1 * G_2$. We write $G_1 * \ldots * G_k$ instead of $((G_1 * G_2) \cdots) * G_k$.

The *split decomposition* of a connected graph is the recursive decomposition of the graph using the simple decomposition until none of the obtained graphs can be decomposed further. Therefore the split decomposition of a graph gives a collection $\mathcal{D}$ of prime graphs. The *split decomposition tree* of the graph G is the tree T in which each node h corresponds to a prime graph G^*_h in $\mathcal{D}$. Furthermore two nodes h and h' in T are adjacent if the corresponding graphs

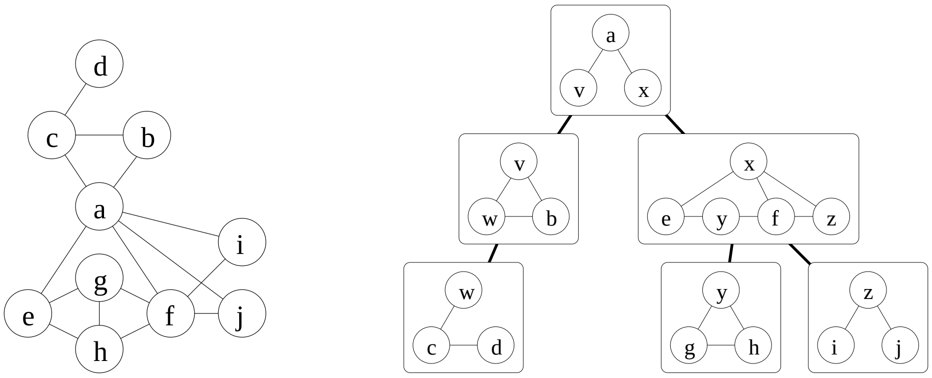

Fig. 2. A graph and its split decomposition tree. The markers are v, w, x, y and z.

G^*_h and $G^*_{h'}$ share a common marker (see figure 2).

With this definition the split decomposition of a graph is not necessarily unique. For instance, K_6 has two non isomorphic decomposition trees. Cunningham [13] showed that every connected graph has an unique decomposition by splits into prime graphs, stars and complete graphs, with a minimum number of components. Our definition corresponds in fact to it's *minimal split decomposition*. It is easy to find a minimal split decomposition from an unique split decomposition, or *vice versa* [13]. Moreover, the set of prime graphs in a minimal decomposition is unique up to isomorphism. From now, every decomposition is supposed to be minimal. The unicity is not needed here.

It is not hard to see that if G is not connected then the parameters we want to compute on G can be easily computed from the parameters of the connected components of G. From now we suppose that the input graph G is connected.

A simple inductive argument shows that a split decomposition tree of a graph G with n vertices and m edges has at most $n-2$ nodes. The sum of the number of vertices of all prime graphs is at most $3n - 4$ since each vertex in a prime graph is either a vertex of G or a marker. The sum of the number of edges of all prime graphs is at most $m + n - 3$ since each simple decomposition adds at most one edge to the overall number of edges.

The first algorithm to compute a split decomposition was given by Cunningham with a running time of $O(n^3)$ [13]. The running time has been improved to $O(nm)$ in [16] and to $O(n^2)$ in [20]. Finally Dahlhaus gave a linear time algorithm in [14].

Some graph classes are nicely decomposable by split decomposition. Distance hereditary graphs are completely decomposable by split decomposition, *i.e.* every prime graph has at most 3 vertices [18]. A graph is a circle graph if and only if every prime graph in its split decomposition is a circle graph [16]. A graph is a parity graph if and only if every prime graph in its split decomposition is bipartite or complete [8]. The best known recognition algorithms for circle graphs [23] and parity graphs [8,14] are based on split decomposition and these forementioned properties. We already know that a graph is perfect if and only if every prime graph in the split decomposition is perfect [2]. A simple decomposition cannot destroy a C_k or its complement $\overline{C_k}$ for some $k \geq 5$. Thus if every prime graph in the split decomposition of G is weakly chordal then G is weakly chordal. Note that is not the case for chordal graphs since C_4 is decomposable.

4 Independent set and clique problems

Cunningham gives an algorithm to compute the weighted stability number using split decomposition [13]. In [8] the authors apply this algorithm to parity graphs using the split decomposition into bipartite and complete graphs. Their algorithm has running time $O(n^{2.5})$. They also present algorithms to compute the clique number and the weighted clique number of a parity graph. Unfortunately these algorithms are flawed. Figure 3 gives a counter-example for the clique number algorithm. This is also a counter-example for the weighted clique number when the weight of every vertex is 1.

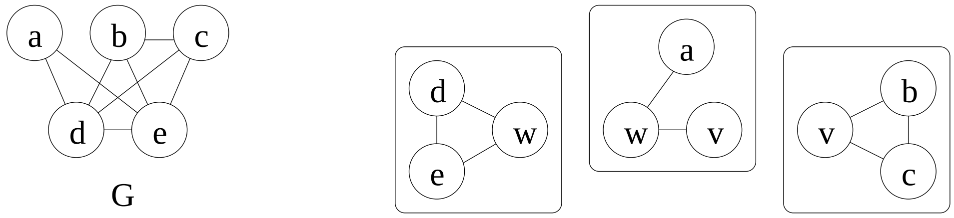

Fig. 3. A parity graph for which the algorithm for the clique number in [8] fails, and its unique split decomposition, which is also its Cunningham decomposition. The markers are v and w. $\omega(G) = 4$ and the output of the algorithm is 3.

Lemma 1 recalls Cunningham's algorithm. Lemma 2 provides a correct algorithm to compute the weighted clique number using the split decomposition.

Let $G = (V, E)$ be a graph such that $G = G_1 * G_2$, $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ and $V_1 \cap V_2 = \{v\}$. Let $w : V \to \mathbb{N}$ be a weight function.

Lemma 1 ([13]) *Let $a = \alpha_w(G_2 - N_{G_2}[v])$ and $a' = \alpha_w(G_2 - v)$. Then*

$$\alpha_w(G) = \alpha_{w|_{v \to a' - a}}(G_1) + a.$$

Lemma 2 *Let $a = \omega_w(G_2[N_{G_2}(v)])$. Then*

$$\omega_w(G) = \max(\omega_w(G_2 - v), \omega_{w|_{v \to a}}(G_1)).$$

PROOF. Let $G' = G[(V_1 \setminus \{v\}) \cup N_{G_2}(v)]$. Suppose that C is a clique of G.

Then either $C \subseteq V_2 \setminus \{v\}$ or $C \subseteq V(G')$. Moreover $\omega_w(G') = \omega_{w|_{v \to a}}(G_1)$ since $N_{G_2}(v)$ is a module in G'. Thus we have $w(C) \leq \max(\omega_w(G_2 - v), \omega_{w|_{v \to a}}(G_1))$. Consequently $\omega_w(G) \leq \max(\omega_w(G_2 - v), \omega_{w|_{v \to a}}(G_1))$.

On the other hand, suppose that C_1 is a clique of G_1, and let C_2 be a clique of maximal weight in G'. If $v \notin C_1$ then C_1 is also a clique in G', and $w|_{v \to a}(C_1) \leq \omega_w(G')$. Otherwise, $C_1 \cup C_2 \setminus \{v\}$ is a clique, and its weight is $w(C_1 \cup C_2 \setminus \{v\}) = w|_{v \to a}(C) \leq \omega_w(G')$, Thus $\omega_{w|_{v \to a}}(G_1) \leq \omega_w(G')$. Obviously, $\omega_w(G) \geq \omega_w(G_2 - v)$ and $\omega_w(G) \geq \omega_w(G')$. Consequently $\omega_w(G) \geq \max(\omega_w(G_2 - v), \omega_{w|_{v \to a}}(G_1))$. $\square$

Lemma 1 (respectively, Lemma 2) can be used to compute $\alpha_w(G)$ (resp. $\omega_w(G)$) for a graph G using the split decomposition tree of G (see [8]). If G is a parity graph, the overall running time for the weighted clique number is linear since the weighted clique number of each component of the split decomposition can be computed in linear time.

5 Domination-type problems

In this section we study the domination problem and some of its variants. We show how to compute the dominating number and its variants using the solutions for the two graphs obtained after a simple split decomposition. It is not hard to see that we can use these lemmas to recursively compute the dominating number and its variants using the split decomposition tree of a graph.

A *dominating set* of G is a subset D of V such that for all $v \in V \setminus D$, there is a $v \in D$ such that $\{u, v\} \in E$. The *weighted domination number*, denoted by $\gamma_w(G)$, is the minimum weight of a dominating set. Some variants of the dominating set have been studied in the literature. We say that D is a *connected dominating set* (resp. a *dominating clique*, an *independent dominating set*, a *total dominating set*) if D is a dominating set and $G[D]$ is connected, (resp. D is a clique, D is a stable set, $G[D]$ has no isolated vertex). Similarly, the *weighted connected domination number* (resp. the *weighted dominating clique number*, the *weighted independent domination number*, the *weighted total dominating number*), denoted by ${\gamma_\mathrm{c}}_w(G)$ (resp. ${\gamma_\mathrm{cl}}_w(G)$, ${\gamma_\mathrm{i}}_w(G)$, ${\gamma_\mathrm{t}}_w(G)$), is the minimum weight of a connected dominating set (resp. of a dominating clique, of an independent dominating set, of a total dominating set of G).

We present in a first time the solution for two variants: the weighted connected dominating set and the weighted dominating clique. For these two problems, it is sufficient to compute the same parameter on G_1 and G_2. In a second time we give a generalization of the weighted dominating set problem, which is in the same time a generalization of the weighted independent dominating set and the weighted total dominating set. We show how to compute the *generalized dominating number* of $G = G_1 * G_2$ with the generalized dominating numbers of G_1 and G_2. That is, if we have a polynomial time algorithm to compute the generalized dominating number on each graph of the decomposition of G, then we can compute the dominating number, the independent dominating number and the total dominating number of G in polynomial time.

5.1 Connected dominating set and dominating clique

Let $G = (V, E)$ be a graph such that $G = G_1 * G_2$, $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ and $V_1 \cap V_2 = \{v\}$. Let $w : V \to \mathbb{N}$ be a weight function.

Lemma 3 *Let $a_1 = \min_{u \in V_1 \setminus \{v\}} w(u)$ and $a_2 = \min_{u \in V_2 \setminus \{v\}} w(u)$. Then the weighted connected domination number of G is:*

$$\gamma_{\mathcal{C}w}(G) = \begin{cases} \min(\gamma_{\mathcal{C}w}(G_1 - v), \gamma_{\mathcal{C}w}(G_2 - v), a_1 + a_2) & \text{if } N_{G_1}[v] = V_1 \text{ and } N_{G_2}[v] = V_2, \\ \gamma_{\mathcal{C}w|_{v \to a_1}}(G_2) & \text{if } N_{G_1}[v] = V_1 \text{ and } N_{G_2}[v] \neq V_2, \\ \gamma_{\mathcal{C}w|_{v \to a_2}}(G_1) & \text{if } N_{G_1}[v] \neq V_1 \text{ and } N_{G_2}[v] = V_2, \\ \gamma_{\mathcal{C}w|_{v \to 0}}(G_1) + \gamma_{\mathcal{C}w|_{v \to 0}}(G_2) & \text{if } N_{G_1}[v] \neq V_1 \text{ and } N_{G_2}[v] \neq V_2. \end{cases}$$

PROOF. In the first case every vertex in $V_1 \setminus \{v\}$ is adjacent to every vertex in $V_2 \setminus \{v\}$. A minimum connected dominating set of G is either a connected dominating set of $G_1 - v$, or a connected dominating set of $G_2 - v$, or has two vertices: one in $V_1 \setminus \{v\}$ and one in $V_2 \setminus \{v\}$. Moreover a connected dominating set of $G_i - v$ (for $i \in \{1, 2\}$) is a connected dominating set of G. Thus we have the equality.

In the second case, let D be a minimum connected dominating set of G. Then $D \cap N_{G_2}(v) \neq \emptyset$ since D is connected, and thus every vertex in $V_1 \setminus \{v\}$ is dominated by a vertex in $D \cap N_{G_2}(v)$. Furthermore every vertex in $V_1 \setminus \{v\}$ has the same neighborhood in $V_2 \setminus \{v\}$. By minimality of D, $|D \cap (V_1 \setminus \{v\})| \leq 1$ and $w(D \cap (V_1 \setminus \{v\})) \in \{0, a_1\}$. Thus $\gamma_{\mathrm{c}w}(G) \leq \gamma_{\mathrm{c}w|_{v \to a_1}}(G_2)$. On the other hand let D' be a connected dominating set of G_2. If $v \notin D'$, then D' is a connected dominating set of G. Otherwise $(D' \setminus \{v\}) \cup \{u\}$ is a connected dominating set of G, where u is a vertex of weight a_1 in $V_1 \setminus \{v\}$. Thus $\gamma_{\mathrm{c}w|_{v \to a_1}}(G_2) \leq \gamma_{\mathrm{c}w}(G)$. The third case is similar to the second.

In the last case, let D be a connected dominating set of G. Note that $D \cap N_{G_i}(v) \neq \emptyset$ (for $i \in \{1, 2\}$) since $G[D]$ is connected. Then $(D \cap V_i) \cup \{v\}$ is a connected dominating set of G_i ($i \in \{1, 2\}$). Thus $\gamma_{\mathrm{c}w}(G) \geq \gamma_{\mathrm{c}w|_{v \to 0}}(G_1) + \gamma_{\mathrm{c}w|_{v \to 0}}(G_2)$. On the other hand, if D_1 is a connected dominating set of G_1 and D_2 is a connected dominating set of G_2, then $(D_1 \cup D_2) \setminus \{v\}$ is connected, is a dominating set of G and $w(D_1 \cup D_2 \setminus \{v\}) = w(D_1) + w(D_2)$. Thus we have $\gamma_{\mathrm{c}_w}(G) = \gamma_{\mathrm{c}_{w|_{v\to 0}}}(G_1) + \gamma_{\mathrm{c}_{w|_{v\to 0}}}(G_2)$. $\square$

Similarly we obtain the following relation for the dominating clique problem. The proof is similar to the proof of Lemma 3 and is omitted.

Lemma 4 *Let $a_1 = \min_{u\in V_1} w(u)$ and $a_2 = \min_{u\in V_2} w(u)$. Then the weighted dominating clique number of G is:*

$$\gamma_{cl_w}(G) = \begin{cases} \min(\gamma_{cl_w}(G_1 - v), \gamma_{cl_w}(G_2 - v), a_1 + a_2) & \text{if } N_{G_1}[v] = V_1 \text{ and } N_{G_2}[v] = V_2, \\ \gamma_{cl_{w|_{v\to a_1}}}(G_2) & \text{if } N_{G_1}[v] = V_1 \text{ and } N_{G_2}[v] \neq V_2, \\ \gamma_{cl_{w|_{v\to a_2}}}(G_1) & \text{if } N_{G_1}[v] \neq V_1 \text{ and } N_{G_2}[v] = V_2, \\ \gamma_{cl_{w|_{v\to 0}}}(G_1) + \gamma_{cl_{w|_{v\to 0}}}(G_2) & \text{if } N_{G_1}[v] \neq V_1 \text{ and } N_{G_2}[v] \neq V_2. \end{cases}$$

5.2 Dominating set, independent dominating set and total dominating set

We define a generalization of the dominating set, the independent dominating set and the total dominating set. Let $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ and $w : V \to \overline{\mathbb{N}}^{\,4}$. For $i \in [\![1, 4]\!]$, let $\Pi_i : \overline{\mathbb{N}}^{\,4} \to \overline{\mathbb{N}}$ be the i-th projection ($i.e.$ $\Pi_i(a_1, a_2, a_3, a_4) = a_i$). Let:

$$\tilde{\mathrm{w}}_w(G, D, u) = \begin{cases} \Pi_1(w(u)) & \text{if } u \in D \text{ and } N_G(u) \cap D \neq \emptyset, \\ \Pi_2(w(u)) & \text{if } u \in D \text{ and } N_G(u) \cap D = \emptyset, \\ \Pi_3(w(u)) & \text{if } u \notin D \text{ and } N_G(u) \cap D \neq \emptyset, \\ \Pi_4(w(u)) & \text{if } u \notin D \text{ and } N_G(u) \cap D = \emptyset. \end{cases}$$

The *generalized weight* of a set D in G is

$$\tilde{\mathrm{w}}_w(G, D) = \sum_{u\in V(G)} \tilde{\mathrm{w}}_w(G, D, u).$$

Let $\gamma_{\mathrm{g}_w}(G)$, *the generalized domination number* of G, be the minimum generalized weight over all subsets of vertices in G. It is easy to see that the generalized domination number is a generalization of the domination number since $\gamma_w(G) = \gamma_{\mathrm{g}_{w'}}(G)$ with $w'(u) = (w(u), w(u), 0, \infty)$ for all $u \in V$. This is also a generalization of the total dominating number with $w'(u) = (w(u), \infty, 0, \infty)$ and the independent dominating number with $w'(u) = (\infty, w(u), 0, \infty)$.

Lemma 5 *Let $G = G_1 * G_2$ with marker v and let:*

$$a_1 = \gamma g_{w|_{v\to(0,\infty,\infty,\infty)}}(G_2),$$
$$a_2 = \gamma g_{w|_{v\to(\infty,\infty,0,\infty)}}(G_2),$$
$$a_3 = \gamma g_{w|_{v\to(\infty,0,\infty,\infty)}}(G_2),$$
$$a_4 = \gamma g_{w|_{v\to(\infty,\infty,\infty,0)}}(G_2).$$

Then

$$\gamma g_w(G) = \gamma g_{w|_{v\to(a_1,a_2,a_3,a_4)}}(G_1).$$

PROOF. Let $V_i' = V_i \setminus \{v\}$, for $i \in \{1,2\}$. $\{V_1, V_2\}$ is a partition of V. Let $w' = w|_{v\to(a_1,a_2,a_3,a_4)}$. Let D be a subset of V and let $D_i = D \cap V_i'$. We show in a first time that $\tilde{\mathrm{w}}_w(G,D) \geq \tilde{\mathrm{w}}_{w'}(G_1,D_1)$ or $\tilde{\mathrm{w}}_w(G,D) \geq \tilde{\mathrm{w}}_{w'}(G_1,D_1\cup\{v\})$.

Suppose that $D \cap N_{G_2}(v) = \emptyset$ and $D \cap N_{G_1}(v) \neq \emptyset$.

$$\begin{aligned}
\tilde{\mathrm{w}}_{w'}(G_1,D_1) &= a_3 + \sum_{u\in V_1'} \tilde{\mathrm{w}}_w(G_1,D_1,u) \qquad \text{since } \tilde{\mathrm{w}}_w(G_1,D,v) = a_3\\
&\leq \tilde{\mathrm{w}}_{w|_{v\to(\infty,0,\infty,\infty)}}(G_2,D_2\cup\{v\}) + \sum_{u\in V_1'} \tilde{\mathrm{w}}_w(G_1,D_1,u)\\
&\leq \sum_{u\in V_2'} \tilde{\mathrm{w}}_w(G_2,D_2\cup\{v\},u) + \sum_{u\in V_1'} \tilde{\mathrm{w}}_w(G_1,D_1,u)\\
&\leq \tilde{\mathrm{w}}_w(G,D).
\end{aligned}$$

Similarly we get (the proofs are omitted):

- if $D \cap N_{G_2}(v) = \emptyset$ and $D \cap N_{G_1}(v) = \emptyset$ then $\tilde{\mathrm{w}}_{w'}(G_1,D_1) \leq \tilde{\mathrm{w}}_w(G,D)$,
- if $D \cap N_{G_2}(v) \neq \emptyset$ and $D \cap N_{G_1}(v) = \emptyset$ then $\tilde{\mathrm{w}}_{w'}(G_1,D_1\cup\{v\}) \leq \tilde{\mathrm{w}}_w(G,D)$,
- if $D \cap N_{G_2}(v) \neq \emptyset$ and $D \cap N_{G_1}(v) \neq \emptyset$ then $\tilde{\mathrm{w}}_{w'}(G_1,D_1\cup\{v\}) \leq \tilde{\mathrm{w}}_w(G,D)$.

In all cases we get $\tilde{\mathrm{w}}_{w'}(G_1,D_1) \leq \tilde{\mathrm{w}}_w(G,D)$ or $\tilde{\mathrm{w}}_{w'}(G_1,D_1\cup\{v\}) \leq \tilde{\mathrm{w}}_w(G,D)$. Thus $\gamma_{\mathrm{g}_{w'}}(G_1) \leq \tilde{\mathrm{w}}_w(G,D)$ and $\gamma_{\mathrm{g}_{w'}}(G_1) \leq \gamma_{\mathrm{g}_w}(G)$.

On the other hand, let D_i be subsets of V_2 (for $i \in [\![1,4]\!]$) such that:

$$\tilde{\mathrm{w}}_{w|_{v\to(0,\infty,\infty,\infty)}}(G_2,D_1) = a_1,$$
$$\tilde{\mathrm{w}}_{w|_{v\to(\infty,\infty,0,\infty)}}(G_2,D_2) = a_2,$$
$$\tilde{\mathrm{w}}_{w|_{v\to(\infty,0,\infty,\infty)}}(G_2,D_3) = a_3,$$
$$\tilde{\mathrm{w}}_{w|_{v\to(\infty,\infty,\infty,0)}}(G_2,D_4) = a_4.$$

For a subset $D' \subseteq V_1 \cup \{v\}$ let:

$$D = \begin{cases} D' \setminus \{v\} \cup D_1 & \text{if } v \in D' \text{ and } D' \cap N_{G_1}(v) \neq \emptyset, \\ D' \setminus \{v\} \cup D_2 & \text{if } v \in D' \text{ and } D' \cap N_{G_1}(v) = \emptyset, \\ D' \cup D_3 & \text{if } v \notin D' \text{ and } D' \cap N_{G_1}(v) \neq \emptyset, \\ D' \cup D_4 & \text{if } v \notin D' \text{ and } D' \cap N_{G_1}(v) = \emptyset. \end{cases}$$

It is not hard to see that $\tilde{\mathrm{w}}_w(G, D) \leq \tilde{\mathrm{w}}_{w'}(G_1, D')$. Thus $\gamma_{\mathrm{g}_w}(G) \leq \tilde{\mathrm{w}}_{w'}(G_1, D')$ and $\gamma_{\mathrm{g}_w}(G) \leq \gamma_{\mathrm{g}_{w'}}(G_1)$. $\square$

6 The coloring problem

In this section we present an algorithm to compute the chromatic number of a graph $G = (V, E)$ using the split decomposition tree of G. We know that we can use the split decomposition to compute an optimal coloring of a perfect graph [2]. But in the general case the problem seems to be more complex, since if $G = G_1 * G_2$, $\chi(G)$ cannot directly be obtained from $\chi_w(G_1)$ and $\chi_w(G_2)$. Typically our algorithm has to compute the weighted chromatic number for a variety of induced subgraphs of G corresponding to nodes of the split decomposition tree.

Let T be a split decomposition tree of the graph G computed using an $O(n^3)$ polynomial time algorithm (for instance Cunningham's algorithm [13] and Gabor *et al.*'s algorithm [16] are easy to implement). Our coloring algorithm runs recursively on the split decomposition tree T in which we choose an arbitrary root r.

We recall that for a node h of T, G^*_h is the prime graph of the split decomposition corresponding to the node h. For each node $h \neq r$ of T and its parent h' in T, let v_h be the unique marker shared by G^*_h and $G^*_{h'}$. For every leaf h

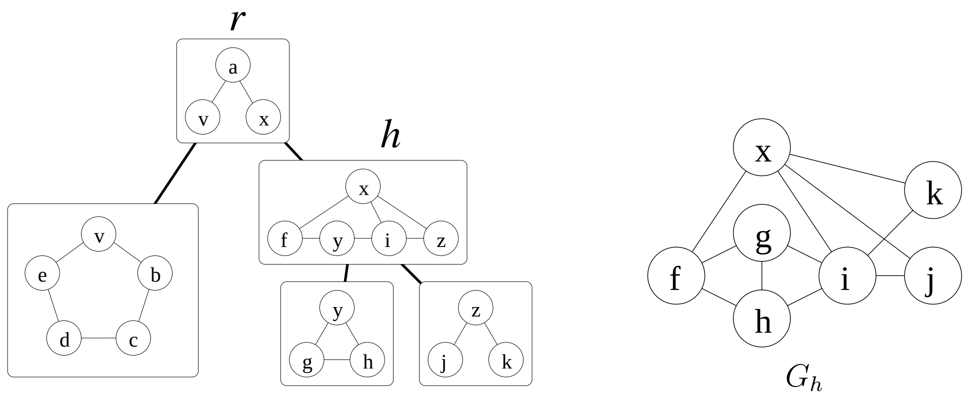

Fig. 4. We take a root r in the split decomposition of our example. ($v_h = x$.)

Function ChromaticNumber(T)
Input: the split decomposition tree T of G
Output: the chromatic number of G
begin
  for each node h of T, in bottom-up fashion
    Let $\{h_1, h_2, \ldots, h_k\}$ be the children of h
    if $h \neq r$ **then**
      $D_h \leftarrow \emptyset$
      for b **from** 0 **to** n
        $D_h \leftarrow D_h \cup \{(\textsc{ChromSplit}(h, b, k, D_{h_1}, D_{h_2}, \ldots, D_{h_k}) - b, b)\}$
      endfor
    else return ChromSplit($r, 0, k, D_{h_1}, D_{h_2}, \ldots, D_{h_k}$)
    endif
  endfor
end

Function ChromSplit($h, b, k, D_{h_1}, D_{h_2}, \ldots, D_{h_k}$)
begin
  if $k = 0$ **then**
    let $w^* : V_h^* \to \mathbb{N}$ such that $\forall v \in V_h^* \setminus \{v_h\}$, $w^*(v) = 1$ and $w(v_h) = b$
    return $\chi_{w^*}(G_h^*)$
  else
    let $m \leftarrow +\infty$
    for c **from** 0 **to** $2n$ **do**
      for i **from** 1 **to** k **do**
        let $a_i \leftarrow \min\{a' : \exists b'$ such that $(a', b') \in D_{h_i}$ and $a' + b' \leq c\}$
      endfor
      let $w^* : V_h^* \to \mathbb{N}$ such that for all $i \in [\![1, k]\!]$, $w^*(v_{h_i}) = a_i$,
        for all $v \in V_h^* \setminus \{v_h, v_{h_1}, \ldots v_{h_k}\}$, $w^*(v) = 1$ and, if $h \neq r$, $w(v_h) = b$
      $m \leftarrow \min(m, \max(\chi_{w^*}(G_h^*), c))$
    endfor
    return m
  endif
end

Fig. 5. Algorithm to compute the chromatic number.

of T, we define $G_h = G_h^*$. We define recursively for each internal node h of T with children $h_1, h_2, \ldots, h_k$, $G_h = G_h^* * G_{h_1} * \ldots * G_{h_k}$. We call G_h the *graph corresponding to the subtree* of T rooted at h. Note that $G_r = G$. An example is given in figure 4.

We present our algorithm in Figure 5. Its correctness proof is based on Lemma 6 and Lemma 7. The following notation is used throughout the proofs of these lemmas. Let C be a w-weighted coloring of $G = (V, E)$ and $V' \subseteq V$. We denote by $C[V']$ the collection of all stable sets in C which have a non-empty intersection with V', *i.e.* $C[V'] = \langle S \in C : S \cap V' \neq \emptyset \rangle$. We denote by $\mathcal{D}(G, w, V')$ the set of all pairs $(a, b) \in \mathbb{N} \times \mathbb{N}$ such that there is a w-

weighted coloring C of G with $a+b$ colors and $|C[V']| = a$. Obviously, if $(a,b) \in \mathcal{D}(G,w,V')$, then $(a',b') \in \mathcal{D}(G,w,V')$ for any $a' \geq a$ and $b' \geq b$. We call a pair $(a,b) \in \mathcal{D}(G,w,V')$ *minimal* if $(a-1,b) \notin \mathcal{D}(G,w,V')$ and $(a,b-1) \notin \mathcal{D}(G,w,V')$. Note that for every minimal pair (a,b) of $\mathcal{D}(G,w,V')$ both $a \leq \chi_w(G)$ and $b \leq \chi_w(G)$.

From now, for every node $h \neq r$, we call $\mathcal{D}(h) = \mathcal{D}(G_h - v_h, w, N_{G_h}(v_h))$ the $\mathcal{D}$-set of h, where for all $v \in V_h \setminus \{v_h\}$, $w(v) = 1$. Our algorithm computes in a bottom-up fashion for all nodes $h \neq r$ the $\mathcal{D}$-set of h from the $\mathcal{D}$-sets of all its children. Finally it computes $\chi(G)$ from the $\mathcal{D}$-sets of the children of the root r.

An example for a $\mathcal{D}$-set is given in figure 6. The graph G_h is the cycle of 5 vertices and every weight is equal to 1. We can color the graph $G_h - v_h$ with 2 colors, but each color class meet the neighborhood of v_h. We can also color the graph with 3 colors, but with only one color meeting $N_{G_h}(v_h)$. Thus $(2,0) \in \mathcal{D}(h)$ and $(1,2) \in \mathcal{D}(h)$.

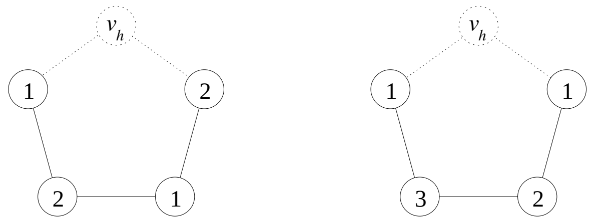

Fig. 6. An example for the $\mathcal{D}$-set when G_h is the C_5. $(2,0) \in \mathcal{D}(h)$ and $(1,2) \in \mathcal{D}(h)$.

The following lemma provides the operation that we execute on each node of the split decomposition tree in order to compute its $\mathcal{D}$-set and its chromatic number using the $\mathcal{D}$-sets of its children. Let h be a node of T. Let $G_h^* = (V_h^*, E_h^*)$ be the prime graph corresponding to h and let $G_h = (V_h, E_h)$ be the graph corresponding to the subtree of T rooted at h. Let $h_1, h_2, \ldots, h_k$ be the children of h in T. Then G_{h_i} is the graph corresponding to the subtree in T rooted at h_i and v_{h_i} is the unique marker shared by G_h^* and G_{h_i}. For simplifying the notations in Lemma 6 and its proof, we note $G_i = G_{h_i} = (V_i, E_i)$, $v_i = v_{h_i}$ and $G^* = G_h^* = (V^*, E^*)$. As discussed above, $G_h = G^* * G_1 * \ldots * G_k$.

Lemma 6 *Let $w : V_h \to \mathbb{N}$ be a weight function. For all $i \in [\![1,k]\!]$, let $(a_i, b_i) \in \mathcal{D}(G_i - v_i, w, N_{G_i}(v_i))$. Let $\mathfrak{C}$ be the set of all w-weighted colorings C of G_h with $|C[V_i \setminus \{v_i\}]| = a_i + b_i$ and $|C[N_{G_i}(v_i)]| = a_i$. Then*

$$\min_{C \in \mathfrak{C}} |C| = \max\left(\chi_{w^*}(G^*), \max_{i \in [\![1,k]\!]} (a_i + b_i)\right)$$

where $w^* : V^* \to \mathbb{N}$ such that for all $i \in [\![1,k]\!], w^*(v_i) = a_i$, and for all $v \in V^* \setminus \{v_1, v_2, \ldots, v_k\}$, $w^*(v) = w(v)$.

PROOF. Recall that if C is a weighted coloring of G and $V' \subseteq V$ then $C[V']$ is defined to be the multiset of all stable sets in C with a non-empty intersection with $V' \subseteq V$. For all $i \in [\![1,k]\!]$, let $A_i = N_{G_i}(v_i)$ and let $B_i = V_i \setminus N_{G_i}[v_i]$.

In a first time, we prove that every coloring C member of $\mathfrak{C}$ contains at least $\max\left(\chi_{w^*}(G^*), \max_{i\in[\![1,k]\!]}(a_i+b_i)\right)$ stable sets. For all $i \in [\![1,k]\!]$, $a_i + b_i = |C[V_i \setminus \{v_i\}]| \leq |C|$. It remains to show that $\chi_{w^*}(G^*) \leq |C|$. To do that we construct a w^*-weighted coloring C^* of G^* from $C[V_h \setminus \bigcup_{i=1}^k B_i]$. For a stable set $S \subseteq V$, let $S^* = S \cap (V^* \setminus \{v_1, \ldots v_k\}) \cup \{v_i : S \cap A_i \neq \emptyset\}$. Note that if S is a stable set of G, then S^* is a stable set of G^*. Let $C^* = \langle S^* \neq \emptyset : S \in C\rangle$. Clearly $|C^*| \leq |C|$. Furthermore C^* is a w^*-weighted coloring of G^* since for all i, $|C^*_{\{v_i\}}| = |C[A_i]| = a_i$.

To show the equality, we construct a w-weighted coloring C of G_h such that $C \in \mathfrak{C}$ and $|C| = \max\left(\chi_{w^*}(G^*), \max_{i\in[\![1,k]\!]}(a_i+b_i)\right)$. Let C^* be a minimum w^*-weighted coloring of G^*, and for all i, let C_i be a coloring of $G_i - v_i$ such that $|C_i| = a_i + b_i$ and $|C_i[A_i]| = a_i$.

We construct C as follows. In the first stage, we pick a stable set $S^* \in C^*$ and for all $i \in [\![1,k]\!]$, if $v_i \in S^*$ we pick a stable set S_i in $C_i[A_i]$, otherwise we pick S_i in $C_i \setminus C_i[A_i]$. If there is no remaining stable set, we take $S_i = \emptyset$. We add to C the set $S = S^* \setminus \{v_1, \ldots v_k\} \cup \bigcup_{i=1}^k S_i$. Note that S is a stable set. We repeat this operation until C^* is empty. Since $w^*(v_i) = a_i = |C_i[A_i]|$, there is no remaining set in $S_i \in C_i$ such that $S_i \cap A_i \neq \emptyset$ at the end of the first stage.

In the second stage, as long as there is an $i \in [\![1,k]\!]$ such that $C_i \neq \emptyset$, we pick for all $i \in [\![1,k]\!]$ a stable set $S_i \in C_i$ (if $C_i = \emptyset$ we take $S_i = \emptyset$). We add to C the set $S = \bigcup_{i=1}^k S_i$. Note that S is a stable set. It is not hard to see that C is a w-weighted coloring of G_h, and $C \in \mathfrak{C}$.

At the end of the first stage, C has $\chi_{w^*}(G^*)$ stable sets, and for each $i \in [\![1,k]\!]$, C_i has $\max(0, b_i - \chi_{w^*}(G^*) + a_i)$ remaining stable sets. Then we add in the second stage $\max\left(0, -\chi_{w^*}(G^*) + \max_{i\in[\![1,k]\!]}(a_i+b_i)\right)$ stable sets to C. At the end the size of C is $\max\left(\chi_{w^*}(G^*), \max_{i\in[\![1,k]\!]}(a_i+b_i)\right)$. $\square$

Let h be a node in T and let $\{h_1, \ldots h_k\}$ be the children of h. Let $w : V_h \to \mathbb{N}$ be a weight function of G_h such that $w(v) = 1$ for all $v \notin V_h^*$. For a k-tuple $\overline{x} = ((a_1,b_1), \ldots, (a_k,b_k))$ into $\mathcal{D}(h_1) \times \ldots \times \mathcal{D}(h_k)$, let $\mathfrak{C}_{\overline{x}}$ be the set of all w-weighted colorings C of G_h with $|C[V_{h_i} \setminus \{v_{h_i}\}]| = a_i + b_i$ and $|C[N_{G_{h_i}}(v_{h_i})]| = a_i$, and let $\chi_{\overline{x}} = \min_{C\in\mathfrak{C}_{\overline{x}}} \max(\chi_{w^*}(G^*), \max_{i\in[\![1,k]\!]}(a_i+b_i))$, where $w^*(v_{h_i}) = a_i$ for all $i \in [\![1,k]\!]$, and $w^*(v) = w(v)$ for all $v \in V_h^* \setminus \{v_{h_1}, \ldots, v_{h_k}\}$. Clearly $\chi_{\overline{x}} \leq \chi_w(G)$. Moreover for every w-coloring C of G_h there is a k-tuple $\overline{x}$ such that $C \in \mathfrak{C}_{\overline{x}}$. Thus by Lemma 6 $\chi_w(G_h) = \min_{\overline{x}\in\mathcal{D}(h_1)\times\ldots\times\mathcal{D}(h_k)} \chi_{\overline{x}}$.

For $c \in \mathbb{N}$, let $\overline{x}_c = ((a_1, b_1), \ldots, (a_k, b_k))$ where for all $i \in [\![1, k]\!]$, (a_i, b_i) is a pair of $\{(a, b) \in \mathcal{D}(h_i) : a + b \leq c\}$ with minimum a (if $\{(a, b) \in \mathcal{D}(h_i) : a + b \leq c\}$ is empty then take $(a_i, b_i) = (+\infty, +\infty)$). Obviously, if $\overline{x}$ is a k-tuple $((a_1, b_1), \ldots, (a_k, b_k))$ and if $c = \max_{i[\![1,k]\!]}(a_i + b_i)$, then $\chi_{\overline{x}_c} \leq \chi_{\overline{x}}$. Thus $\chi_w(G_h) = \min_{c \in [\![0,2n]\!]} \chi_{\overline{x}_c}$. The algorithm in Figure 5 computes $\chi_w(G_h)$ by taking the minimum over all k-tuples t_c, for all $c \in [\![0, 2n]\!]$.

We known now how to compute the weighted chromatic number of G_h from the $\mathcal{D}$-set of its children. If h is a node different from r, then we have to compute the $\mathcal{D}$-set of h. The following lemma shows how this can be done.

Lemma 7 *Let $G = (V, E)$ be a graph, let $v \in V$, let $w : V \setminus \{v\} \to \mathbb{N}$ be a weight function, and let $b \in \mathbb{N}$. Then $(\chi_{w|_{v \to b}}(G) - b, b) \in \mathcal{D}(G - v, w, N(v))$ and $(\chi_{w|_{v \to b}}(G) - b - 1, b) \notin \mathcal{D}(G - v, w, N(v))$.*

PROOF. Let $w' = w|_{v \to b}$. Let C be a minimum w'-weighted coloring of G. Then $C' = \langle S \setminus \{v\} : S \in C \rangle$ is a w-weighted coloring of G and has at least b stable sets with no intersection with $N(v)$. Thus $(\chi_{w'}(G) - b, b) \in \mathcal{D}(G - v, w, N(v))$.

Suppose that there is a w-weighted coloring $\widehat{C}$ of $G - v$ such that $|\widehat{C}| < \chi_{w'}(G)$ and $|\widehat{C} \setminus \widehat{C}[N(v)]| \geq b$. Then the multiset C obtained by adding v to every stable set in $\widehat{C} \setminus \widehat{C}[N(v)]$ will be a w'-weighted coloring of G. Contradiction. $\square$

Lemma 7 says that if we know $\chi_{w|_{v_h \to b}}(G_h)$ for every $b \in [\![0, n]\!]$, then we can compute the $\mathcal{D}$-set of h in linear time. Now we are ready to summarize the correctness proof.

Theorem 8 *The algorithm* ChromaticNumber *takes as input a graph G and its split decomposition tree T and computes the chromatic number of G.*

PROOF. Let h be a node, and $\{h_1, \ldots h_k\}$ be its children in T rooted at r. We suppose by induction than D_{h_i} contains all the minimal pairs of the $\mathcal{D}$-set of h_i. By Lemma 6 and the remark following its proof, the function ChromSplit$(h, b, \ldots)$ returns $\chi_w(G_h)$, with $w(v) = 1$ for all v in $V_h \setminus \{v_h\}$ and $w(v_h) = b$ (if v_h exists). If $h \neq r$, by Lemma 7, the main loop computes D_h which contains all the minimal pairs of the $\mathcal{D}$-set of h. Otherwise, the algorithm computes the chromatic number of $G_r = G$. $\square$

Our algorithm computes the chromatic number of the input graph. We mention that it is not hard to modify the algorithm such that it computes a minimum coloring if we use as sub-function an algorithm which computes a minimal weighted coloring of a prime graph.

Theorem 9 *If the sub-algorithm for weighted chromatic number for the prime graphs in the input split decomposition has polynomial running time $f(n,m)$ then the total running time of the algorithm* ChromaticNumber *is $O(n^3 \cdot f(n,m))$. Moreover if $f(n_1+n_2, m_1+m_2) \leq f(n_1,m_1)+f(n_2,m_2)$ for any positive integers n_1, n_2, m_1 and m_2, then the total running time is $O(\max(n^3, n^2 \cdot f(n,m)))$.*

PROOF. This algorithm executes $O(n^2)$ times the sub-algorithm for the weighted chromatic number for each prime graph in the split decomposition, and the split decomposition tree has $O(n)$ nodes. The remainder can be done in time $O(n^3)$ by pre-calculating $\min\{a' : \exists b'$ such that $(a',b') \in D_h$ and $a'+b' \leq c\}$ for every node h in T and for every $c \in [\![0, 2n]\!]$. Consequently the overall running time is $O(n^3 \cdot f(n,m))$.

Moreover if $f(n_1+n_2, m_1+m_2) \leq f(n_1,m_1) + f(n_2,m_2)$ the time needed to run the sub-algorithm on each prime graph is at most $f(\sum_i n_1, \sum_i m_i) = O(f(n,m))$. Thus the total running time is $O(\max(n^3, n^2 \cdot f(n,m)))$. $\square$

7 Clique-width

In this section we show that if the clique-width of the prime graphs in the split decomposition of G is bounded by k, then the clique-width of G is bounded by $2k+1$. Such a theorem is already known for the modular decomposition.

A *k-labeled graph* is graph for which every vertex has a label $l(v)$ in $[\![1,k]\!]$. It is denoted by (V,E,l). We denote by $l(G)$ the labeling function of a labeled graph G. Let the following operations:

(1) (Create a vertex) For $i \in [\![1,k]\!]$, $(v)_i$ is the graph with one vertex v labeled i.
(2) (Disjoint union) Let $G_1 = (V_1, E_1, l_1)$ and $G_2 = (V_2, E_2, l_2)$ be two k-labeled graphs. We suppose that $V_1 \cap V_2 = \emptyset$ (otherwise take a copy of G_2). Then $G_1 \oplus G_2$ is the k-labeled graph $(V_1 \cup V_2, E_1 \cup E_2, l)$ where for all $v \in V_1 \cup V_2$:
$$l(v) = \begin{cases} l_1(v) \text{ if } v \in V_1, \\ l_2(v) \text{ if } v \in V_2. \end{cases}$$
(3) (Relabel) Let $G = (V,E,l)$ be a k-labeled graph and let $i,j \in [\![1,k]\!]$ with $i \neq j$. Then $\rho_{i \to j}(G)$ is the k-labeled graph (V,E,l') where $l'(v) = l(v)$ if $l(v) \neq i$, and $l'(v) = j$ if $l(v) = i$.

(4) (Create edges) Let $G = (V, E, l)$ be a k-labeled graph and let $i, j \in [\![1, k]\!]$ with $i \neq j$. Then $\eta_{(i,j)}(G)$ is the k-labeled graph $(V, E \cup E', l)$ where $E' = \{\{u, v\} : (u, v) \in V^2 \text{ and } (l(u), l(v)) = (i, j)\}$.

A *k-expression* is an expression built with $(v)_i$, $\oplus$, $\eta_{(i,j)}$ and $\rho_{i\to j}$, with labels into $[\![1, k]\!]$. We denote by G^t the labeled graph defined by the expression t. The *clique-width* [9] of a labeled graph G (denoted by $\mathrm{cwd}(G)$) is the smallest integer k such that there is a k-expression t and $G^t = G$. The clique-width of an unlabeled graph G is the minimum of the clique-width over all labellings of vertices of G, and a k-expression for G is a k-expression for a labeling of G.

Theorem 10 ([11]) *For every graph G, $\mathrm{cwd}(G) = \max\{\mathrm{cwd}(H) : H$ is a representative graph of an internal node in the modular decomposition of $G\}$.*

We present an analogue theorem for the split decomposition.

Theorem 11 *Let G be a graph, and let $k = \max\{\mathrm{cwd}(H) : H$ is a representative graph of an internal node in the split decomposition of $G\}$. Then $k \leq \mathrm{cwd}(G) \leq 2k + 1$.*

PROOF. Clearly $k \leq \mathrm{cwd}(G)$ since every representative graph in the split decomposition of G is a subgraph of G. Let T be the split decomposition tree of G, and for all node h in T, let t^*_h be a k-expression for G^*_h. We choose an arbitrary root r in T. We use here the notations v_h, G_h and V_h introduced in Section 6. We build recursively a $(2k+1)$-expression t_h for $G_h - v_h$ such that for all $v \in N_{G_h}(v_h)$, $l(G^{t_h})(v) = 1$ and for all $v \notin N_{G_h}(v_h)$, $l(G^{t_h})(v) = 2k + 1$.

Let h be a node of T and let $\{h_1, \ldots, h_q\}$ be the children of h. By induction for all $i \in [\![1, q]\!]$ we have a $(2k+1)$-expression t_{h_i} for $G_{h_i} - v_{h_i}$. In a first step (if $h \neq r$) we construct a $2k$-expression t'_h for $G^*_h - v_h$ such that for all $v \in N_{G^*_h}(v_h)$, $l(G^{t'_h})(v) = l(G^{t_h})(v)$, otherwise $l(G^{t'_h})(v) = l(G^{t_h})(v) + k$. For that, we substitute in t^*_h the sub-expression $(v)_i$ by $(v)_{i+k}$ for every $v \notin N_{G^*_h}(v_h)$. Moreover we substitute each $\eta_{(i,j)}$ by $\eta_{(i,j)} \circ \eta_{(i+k,j+k)}$, and each $\rho_{i\to j}$ by $\rho_{i\to j} \circ \rho_{(i+k)\to(j+k)}$. (If $h = r$ then we take $t'_r = t^*_r$.)

Now we construct t_h from t'_h as follows. For all $i \in [\![1, q]\!]$, we substitute in t'_h the sub-expression $(v_{h_i})_l$ by $\rho_{1\to l}(t_{h_i})$. Furthermore we add $\circ_{i\in[\![2,k]\!]}\rho_{i\to 1} \circ \circ_{i\in[\![k+1,2k]\!]}\rho_{i\to 2k+1}$ on the top of the expression (*i.e.* we relabel every vertex with label into $[\![k+1, 2k]\!]$ with the label $2k+1$, and every vertex with label into $[\![2, k]\!]$ with the label 1).

We show that t_h is a $(2k+1)$-expression for $G_h - v_h$ (or for G_h if $h = r$). For $i \in [\![1, q]\!]$, let $A_i = N_{G_{h_i}}(v_{h_i})$ and let $B_i = V_{h_i} \backslash N_{G_{h_i}}[v_{h_i}]$. Let $C = V^*_h \backslash \{v_1, \ldots, v_q\}$. Note that $\{C, A_1, \ldots, A_q, B_1, \ldots, B_q\}$ is a partition of V_h. The clique-width operations do not allow to add an edge between two vertices with the same label. Moreover we never add an edge between a vertex labeled $2k+1$ and an other vertex, so for all $i \in [\![1,q]\!]$, $G^{t_h}[A_i \cup B_i] = G^{t_{h_i}} = G_{h_i} - v_{h_i}$. Furthermore we never add an edge between a vertex in B_i and a vertex in $C \cup \bigcup_{j \neq i} B_j$. For all $i \in [\![1,q]\!]$, A_i is a module in $G_h - \bigcup_{i \in [\![1,q]\!]} B_i$. Every vertex in A_i has the same label in t_h, thus A_i is also a module in $G^{t_h} - \bigcup_{i \in [\![1,q]\!]} B_i$. Moreover, by the construction of t_h, for every $v \in A_i$, $N_{G^{t_h}}(v) \setminus (A_i \cup B_i) = N_{G_h}(v) \setminus (A_i \cup B_i)$. Thus $G^{t_h} - \bigcup_{i \in [\![1,q]\!]} B_i = G_h - v_h - \bigcup_{i \in [\![1,q]\!]} B_i$. To summarize, we get $G^{t_h} = G - v_h$, and $G_r = G$. $\square$

8 Polynomial time algorithms

To conclude, we present some graph classes on which previous algorithms can be applied. In order to design a polynomial time algorithm for a problem Π, we have to find a class of graphs $\mathcal{G}$ such that (1) the closure of graphs in $\mathcal{G}$ by the simple composition is a proper super-set of $\mathcal{G}$ and (2) we can solve in polynomial time the weighted version of Π on $\mathcal{G}$. We give two examples of such graph classes.

8.1 Every prime graph has bounded size

Let $\mathcal{G}_k$, for $k \geq 3$, be the class of graph for which every prime graph in a split decomposition has at most k vertices. Note that $\mathcal{G}_3$ is exactly the class of distance hereditary graphs. By Theorem 11, the clique-width of the graph class $\mathcal{G}_k$ is bounded by $2k+1$. For graphs of size bounded by a fixed k, the weighted stability number, the weighted clique number, the weighted connected domination number, the weighted dominating clique number and the generalized domination number can be computed in constant time, since there is a fixed number of subsets of vertices. Moreover the weighted chromatic number of graphs of bounded size can be computed in constant time [3].

Corollary 12 *For any fixed $k \geq 3$, there are $O(n)$ algorithms to compute the weighted stability number, the weighted clique number, the domination number and its variants, and an $O(n^3)$ algorithm to compute the chromatic number of graph in the class $\mathcal{G}_k$, if a split decomposition is given with the graph.*

8.2 Every prime graph is an induced subgraph of C_k or $\overline{C_k}$ for some $k \geq 3$

Let $\mathcal{G}_c$ be the class of graphs for which every prime graph in the split decomposition is an induced subgraph of C_k or $\overline{C_k}$, $k \geq 3$. We can easily see that the clique-width of a C_k or $\overline{C_k}$ is bounded by 4. By Theorem 11, the clique-width of this class is bounded by 9. For all these graphs, computing the weighted stability number, the weighted clique number, the weighted connected domination number, the weighted dominating clique number and the generalized domination number can be computed in linear time by dynamic programming. Thus using results of Sections 4 and Section 5 we obtain:

Corollary 13 *There are $O(n+m)$ algorithms to compute the weighted stability number, the weighted clique number, the domination number and its variants on the graph class $\mathcal{G}_c$.*

A more challenging problem is to compute the chromatic number.

Lemma 14 *Let $G = (V, E)$ with $V = \{v_1, v_2, \dots, v_n\}$ be a cycle of length $n \geq 3$, and let $w : V \to \mathbb{N}$ be a weight function of G. Then*

$$\chi_w(G) = \max\left(\omega_w(G), \left\lceil \frac{\sum_{i=1}^n w(v_i)}{\left\lfloor \frac{n}{2} \right\rfloor} \right\rceil\right).$$

PROOF. Let $W = \sum_{i=1}^n w(v_i)$ and let $p = \lfloor n/2 \rfloor$. Clearly we have $\chi_w(G) \geq \omega_w(G)$. Furthermore each color class in a coloration of (G, w) has at most p vertices, and thus $\chi_w(G) \geq \lceil W/p \rceil$. If n is even or $n = 3$ then G is perfect, and thus $\chi_w(G) = \omega_w(G)$ [17].

Now suppose that n is odd and at is least 5. We show the claimed equality by induction on W. If there is a $v \in V$ such that $w(v) = 0$, then $\chi_w(G) = \chi_w(G - v) = \omega_w(G)$ since $G - v$ is perfect. Otherwise let E' be the set of edges $\{a, b\} \in E$ such that $w(a) + w(b) = \omega_w(G)$.

Case 1: $E' = E$. All weights are equals since n is odd, thus $\lceil W/p \rceil > \omega_w(G)$. Let S be a stable set with p vertices, and let w' be the weight function defined by $w'(v) = w(v) - 1$ for all $v \in S$ and $w'(v) = w(v)$ for all $v \in V \setminus S$. Let $W' = \sum_{i=1}^n w'(v_i)$. Then

$$\begin{aligned}
\chi_w(G) &\leq 1 + \chi_{w'}(G) \\
&\leq 1 + \max\left(\omega_{w'}(G), \lceil W'/p \rceil\right) \text{ (by induction)} \\
&\leq \max\left(\omega_w(G), \lceil W/p \rceil\right)
\end{aligned}$$

since $\lceil W/p \rceil > \omega_w(G) = \omega_{w'}(G)$ and $\lceil W'/p \rceil = \lceil W/p \rceil - 1$.

Case 2: $E' \neq E$. Then there is a stable set S of p vertices containing an endpoint of each edge in E'. Let w' be the weight function defined by $w'(v) = w(v) - 1$ for all $v \in S$ and $w'(v) = w(v)$ for all $v \in V \setminus S$. Let $W' = \sum_{i=1}^n w'(v_i)$.

Then

$$\begin{aligned}\chi_w(G) &\leq 1 + \chi_{w'}(G) \\ &\leq 1 + \max\left(\omega_{w'}(G), \lceil W'/p \rceil\right) \text{ (by induction)} \\ &\leq \max\left(\omega_w(G), \lceil W/p \rceil\right)\end{aligned}$$

since $\lceil W'/p \rceil = \lceil W/p \rceil - 1$ and $\omega_{w'}(G) = \omega_w(G) - 1$. $\square$

The *weighted clique covering number* of a weighted graph (G, w) is $\kappa_w(G) = \chi_w(\overline{G})$. The following lemma gives the weighted clique partition number of a graph C_n for $n \geq 3$. Thus it gives the weighted chromatic number of a graph $\overline{C_n}$ for $n \geq 3$. The proof is similar to the proof of Lemma 14 and is omitted.

Lemma 15 *Let $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$ be a cycle of length $n \geq 3$, and let $w : V \to \mathbb{N}$ be a weight function of G. Then*

$$\kappa_w(G) = \max\left(\alpha_w(G), \left\lceil \frac{\sum_{i=1}^{n} w(v_i)}{2} \right\rceil\right).$$

By Lemma 14 and Lemma 15 the weighted chromatic number of a graph C_k or $\overline{C_k}$, $k \geq 3$, can be computed in linear time. Using Theorem 9 we obtain:

Corollary 16 *There is an $O(n^2 m)$ algorithm to compute the chromatic number for graphs in the class $\mathcal{G}_c$.*

Acknowledgement

I am grateful to D. Kratsch and to anonymous referees for helpful comments.

References

[1] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, *Journal of Algorithms* 12(2) (1991) 308–340.

[2] R. Bixby, A composition for perfect graphs, *Annals of Discrete Math.* 21 (1984) 221–224.

[3] H. L. Bodlaender, A. Brandstädt, D. Kratsch, M. Rao, J. Spinrad, On algorithms for (P_5,gem)-free graphs, *Theoretical Computer Science*, 349(1) (2005) 2–21.

[4] H.L. Bodlaender, K. Jansen, On the complexity of the maximum cut problem, *Nord. J. Comput.* 7(1) (2000) 14–31.

[5] H. L. Bodlaender, U. Rotics, Computing the treewidth and the minimum fill-in with the modular decomposition, *Algorithmica* 36(4) (2003) 375–408.

[6] A. Brandstädt, D. Kratsch, On the structure of (P_5,gem)-free graphs, *Discrete Applied Math.* 145(2) (2005) 155–166.

[7] M. Chudnovsky, N. Robertson, P. D. Seymour, R. Thomas, The strong perfect graph theorem, *Annals of Math.* 164(1) (2006) 51–229.

[8] S. Cicerone, D. Di Stefano, On the extension of bipartite graphs to parity graphs, *Discrete Applied Math.* 95(1-3) (1999) 181–195.

[9] B. Courcelle, J. Engelfriet, G.Rozenberg, Handle-rewriting hypergraph grammars, *Journal of Computer and System Sciences* 46(2) (1993) 218–270.

[10] B. Courcelle, J. A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory of Computing Systems* 33(2) (2000) 125-150.

[11] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, *Discrete Applied Math.* 101(1-3) (2000) 77-114.

[12] A. Cournier, M. Habib, A new linear algorithm for modular decomposition, *Proceedings of Trees in Algebra and Programming - CAAP'94*, LNCS 787 (Springer-Verlag, Berlin, 1994) 68–84.

[13] W. Cunningham, Decomposition of directed graphs, *SIAM Journal on Algebraic and Discrete Methods* 3(2) (1982) 214–228.

[14] E. Dahlhaus, Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition, *Journal of Algorithms* 36(2) (2000) 205-240.

[15] W. Espelage, F. Gurski, E. Wanke, How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time, *Proceedings of WG 2001*, LNCS 2204 (Springer-Verlag, Berlin, 2001) 117–128.

[16] C. P. Gabor, W. L. Hsu, K. J. Supowit, Recognizing circle graphs in polynomial time, *Journal of the ACM* 36(3) (1989) 435–473.

[17] M. Grötschel, L. Lovász, A. Schrijver, Polynomial algorithms for perfect graphs, *Annals of Discrete Math.* 21 (1984) 325–356.

[18] P. Hammer, F. Maffray, Completely separable graphs, *Discrete Applied Math.* 27(1-2) (1990) 85–99.

[19] D. Kobler, U. Rotics, Edge dominating set and colorings on graphs with fixed clique-width, *Discrete Applied Math.* 126(2-3) (2003) 197–221.

[20] T.-H. Ma, J. Spinrad, An $O(n^2)$ algorithm for undirected split decomposition, *Journal of Algorithms* 16(1) (1994) 154–160.

[21] R.M. McConnell, J. Spinrad, Modular decomposition and transitive orientation, *Discrete Math.* 201(1-3) (1999) 189–241.

[22] R.H Möhring, F.J. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization, *Annals of Discrete Math.* 19 (1984) 257–356.

[23] J. Spinrad, Recognition of circle graphs, *Journal of Algorithms* 16(2) (1994) 264–282.

[24] R. E. Tarjan, Decomposition by clique separators, *Discrete Math.* 55 (1985) 221-232.

[25] M. Tedder, D. Corneil, M. Habib, C. Paul, Simple, Linear-time Modular Decomposition, *Manuscript (2007).* http://arxiv.org/abs/0710.3901