SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'Informazione
Informatica e Statistica
Dipartimento di Scienze Statistiche

DOTTORATO DI RICERCA IN RICERCA OPERATIVA - XXIV ciclo

On the clique cover problem on claw-free perfect graphs

Candidato
Claudia Snels

Relatore
Prof. Gianpaolo Oriolo

Marzo 2012

Alla mia Famiglia, quella vecchia e quella nuova

Alice nel paese delle meraviglie

"Ma io non voglio andare fra i matti", osservò Alice.

"Be', non hai altra scelta", disse il Gatto "Qui siamo tutti matti. Io sono matto. Tu sei matta."

"Come lo sai che sono matta?" disse Alice.

"Per forza," disse il Gatto: "altrimenti non saresti venuta qui."

Lewis Carroll

Contents

List of Figures

Introduction

The main goal of this thesis is to develop new combinatorial algorithms for the clique cover problem (weighted and unweighted) on perfect graphs. This problem has not received the same interest of its dual problem, the maximum stable set problem; the most recent results on the minimum clique cover problem on perfect graphs go back to the Eighties (Groetschel, Lovàsz, and Schrijver in [21] and Hsu and Nemhauser in [22] and [23] for the subclass of claw-free perfect graphs).

In the last years a lot of efforts have been devoted to a better understanding of the structure of perfect graphs, mainly in trying to prove the strong perfect graph conjecture (that was finally proved by Chudnovsky, Robertson, Seymour and Thomas in [7]), and of other relevant classes of graphs, such as claw-free graphs (with an outstanding series of papers by Chudnovsky and Seymour, for a survey see [8]). These results introduce the idea that a graph can be obtained with a composition of simpler graphs, called strips. The understanding of the structure of claw-free graphs together with this new composition operator for graphs has been the key for the development of a new combinatorial algorithm for the maximum weighted stable set problem (see [34] and [17]).

We want to exploit all the acquired knowledge on the structure of perfect and claw-free graphs as it has already been done for the maximum weighted stable set, to produce new algorithms for the minimum (weighted and unweighted) clique cover problem.

Let us now summarize the outcomes of our work, together with the way those are organized in this thesis. We start in Chapter 1 with general definitions and some basic properties and results on the combinatorial problems and classes of graphs mainly treated in this thesis. Then we move to the minimum clique cover problem on claw-free perfect graphs; we present in Chapter 2 a combinatorial algorithm for this problem which runs in $O(|V|^3)$ and builds concurrently a minimum clique cover and a maximum stable set of the graph. In Chapter 3 we study the weighted version of the problem and we present an algorithmic theorem for the minimum weighted clique cover problem on strip composed perfect graphs. This result will be one of the building blocks for the algorithm for the minimum weighted clique cover on strip-composed claw-free perfect graphs presented in Chapter 4.

In the second part of the thesis (Chapter 5) we present a reduction technique to remove proper and homogeneous pairs of cliques from a graph while preserving some graph invariants. For some classical discrete optimization problems, especially in claw-free graphs, proper and homogeneous pairs of cliques represent an 'annoying' configuration of vertices, thus some preprocessing routines have been developed in the literature, that eliminate proper and homogeneous pairs of cliques. For example a reduction of proper and homogeneous pairs of cliques for the MWSS in claw-free graphs is presented in [34], and for the maximum clique and coloring problem in quasi-line graphs in [24]. In the thesis we introduce a *family* of reductions that can be used for removing proper and homogeneous pairs of cliques from a graph G while maintaining some given graph invariant. This family includes the routines presented in the literature, underlining the common framework behind them. Our reductions can be embedded in a simple algorithm that in at most $|E(G)|$ steps builds a new graph G' without proper and homogeneous pairs of cliques, and such that G and G' agree on the value of the chosen invariant.

Sources

Results in Chapters 2, 3 and 4 are joint work with Gianpaolo Oriolo and Flavia Bonomo; an extended abstract of the results in Chapters 3 and 4 appears in the proceedings of the 2011 Cologne Twente Workshop. Results in Chapter 5 are joint work with Gianpaolo Oriolo and Yuri Faenza and are published in [16].

Chapter 1

Basic notations and preliminary notions

We begin with a chapter on general purpose notations and notions that will be used throughout the thesis. The exposition is very far from being exhaustive: for any background material that we missed and a wider exposition of the topics presented, the reader may refer to [44] for graphs, to [41] for combinatorial optimization and to [40] for polyhedra, linear and integer programming. We start with some general notations and definitions.

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ denote respectively the set of natural, integer, rational and real numbers. Given $n \in \mathbb{N}$, we denote by $[n]$ the finite set $\{1, 2, \ldots, n\}$. By $\mathbb{Q}^+$ (resp. $\mathbb{R}^+$) we denote the set of rational (resp. real) non-negative numbers. Let $f, g : \mathbb{N} \to \mathbb{R}$ be functions from the set of natural numbers to set of real numbers. We say that $f = \mathrm{O}(g)$ if there exist constants c and N such that $f(n) \leq c \cdot g(n)$ for all integers $n \geq N$. Given a set S, we define 2^S to be the set of all subsets of S; if moreover S has a finite number of elements, the *size* or *cardinality* $|S|$ of S is the number of distinct elements S contains. If $f : S \to \mathbb{R}$, for any $\bar{S} \subseteq S$ we define $f(\bar{S}) := \sum_{s \in \bar{S}} f(s)$.

1.1 Graphs

An *undirected graph* is an ordered pair $G := (V, E)$, where V is a set of *vertices* and E is a set of unordered pairs of vertices each of which is called *edge*. Alternatively, we denote respectively by $V(G)$ and $E(G)$ the set of vertices and the set of edges of G. With a slight abuse of notation, we denote by (u,v) or uv the edge corresponding to the unordered pair $\{u,v\}$. If $(u,v) \in E$, we say that u,v are the *extremes* or *endpoints* of edge (u,v), and that u,v are *adjacent* or *joined by an edge* in G. If there is a pair of vertices $\{u,v\}$ occurring more than once in E, we say that this pair is a *multiple* edge. We deal with *loopless* graphs, i.e. we assume that $(u,u) \notin E$ for each $u \in V$. For each $v \in V$, we denote by $\delta(v)$ the set of edges of G with an endpoint in v. We mostly deal with *simple* graphs, that is graphs without multiple edges; when our graph G will not be simple we will call it *multigraph*. Given a set $U \subseteq V$, we denote by $E(U)$ the set of edges of G with exactly one endpoint in U.

A *subgraph* of a graph $G(V,E)$ is a graph $G'(V',E')$ of G with $E' \subseteq E$ and $V' \subseteq V$ and $u,v \in V'$ for each $(u,v) \in E'$. G' is an *induced* subgraph of G if, moreover, $(u,v) \in E'$ if and only if $u,v \in V'$ and $(u,v) \in E$. Thus an induced subgraph is uniquely identified by a set $V' \subseteq V$, and we denote it by $G[V']$. Sometimes we shall refer to $G \setminus V'$ as to the subgraph of G induced by $V \setminus V'$. Given a graph $G(V,E)$ and an integer $k \in \mathbb{N}$, an ordered set of vertices $v_1,\dots,v_k$ is a *walk* (*of length* $k-1$) if $(v_i,v_{i+1}) \in E$ for each $i \in [k-1]$. If $v_1,\dots,v_k$ are all distinct, the walk is called a *path*. If $P = v_1,\dots,v_k$ is a path, the vertex v_1 is called the *starting vertex* or *first vertex* of P and the vertex v_k the *end vertex* or *last vertex* of P. Sometimes both v_1 and v_k are called the *end vertices* or *extremes* or *ends* of P. We say that a path $P = v_1,\dots,v_k$ is *induced* if, moreover, for every $1 \le i < j$ with $j \neq i+1$, $v_iv_j \notin E$. A graph is *connected* if there exists a walk between any two vertices of G. For $j \in \mathbb{N}$, the j-th *neighborhood* $N_j(v)$ of a vertex $v \in V$ is the set of vertices $u \in V$ such that the minimum length of a walk joining v and u is j (the graph G will be clear from the context). In particular, the (*first*) *neighborhood* of v in G ($N_1(v)$ or $N(v)$) is the set of vertices that are joined to v by an edge. We will denote with $N[v]$ the *closed neighborhood* of v, that is $N[v] := N(v) \cup \{v\}$. A vertex v is *isolated* in G if $N(v) = \emptyset$. A *connected component* of G is a maximal (w.r.t. to the vertex set) connected induced subgraph of G. Two graphs G and H are said to be *isomorphic* if there exists a bijection $\phi : V(G) \to V(H)$ such that $(u, v) \in E(G)$ if and only if $(\phi(u), \phi(v)) \in E(H)$. Given graphs G and H, we say that G *contains* H if there exists an induced subgraph of G that is isomorphic to H. A graph that does not contain any induced subgraph isomorphic to a given graph H is said to be H-free. The *complement* of a graph G is the graph $\overline{G}(V, \overline{E})$ where $\overline{E}$ is the set of edges (u, v) with $u \neq v$ such that $(u, v) \notin E$. A graph G is *complete* (or *a complete graph*) if its complement has no edge. Sometimes, especially when dealing with induced subgraphs, we shall refer to an *anti-G* as the complement of G. Given $U, U' \subset V$, we say that U, U' are *complete* (*to each other*) in $G(V, E)$ if for each $u \in U$, $u' \in U'$, $(u, u') \in E$. We say they are *anticomplete* (*to each other*) in $G(V, E)$ if they are complete in the complement of G. For some $n \in \mathbb{N}$, an *n-hole* is a graph with n vertices $u_1, \ldots, u_n$ and edges (u_i, u_{i+1}) for each $i \in [n-1]$ plus the edge (u_n, u_1). An *odd hole* is an n-hole with $n \geq 4$ odd. Similarly, one defines even holes, odd antiholes, even antiholes.

Given a graph $G(V, E)$, we say that a set $U \subseteq V$ is a *stable set* of G if no two elements of its are joined by an edge in G, while it is a *clique* if each two elements of it are joined by an edge in G. We denote by $\alpha(G)$ the size of the maximum stable set in G, and we often refer to $\alpha(G)$ as to the *stability number* of G. We denote by $\omega(G)$ the size of the maximum clique in G, and we often refer to $\omega(G)$ as to the *clique number* of G. A *coloring* of a graph $G(V, E)$ is a function $f : V \to \mathbb{N}$ with the property that $f(u) \neq f(v)$ for each $u, v \in V$ with $(u, v) \in E$. The *chromatic number* of G, denoted by $\chi(G)$, is the size of the smallest co-domain over all functions f that are colorings of G. Equivalently (but in a more combinatorial fashion), a coloring is a function that assigns to each vertex of G a color such that two adjacent vertices of G are not given the same color. The chromatic number of G is then the cardinality of the smallest set C of colors such that there exists a coloring of G that uses only colors from C. We say that G is *k-colorable* if there exists a coloring of G that uses only colors from C, with $|C| = k$.

A graph $G(V, E)$ is *k-partite* if V can be partitioned in k sets $V_1, \ldots, V_k$ and each edge of G has an endpoint in V_i and one in V_j with $i \neq j$ (i.e. V_i is a stable set for every $i = 1, \ldots, k$). In the special case $k = 2$ the graph is said to be *bipartite*. We will often refer to complements of bipartite graphs as *cobipartite* graphs: in those graphs the vertex set can be covered with two cliques.

The intersection graph of a family of sets $\mathcal{C}$ is the graph with vertex set $\mathcal{C}$, two sets in $\mathcal{C}$ being adjacent if and only if they intersect.

1.2 Claw-free graphs

A graph is *claw-free* if it does not contain any induced subgraph isomorphic to a claw (pictured in fig. 1.1).

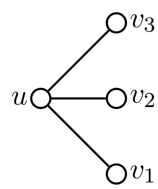


Figure 1.1: A claw $(u; v_1, v_2, v_3)$

Claw-free graphs play a relevant role in combinatorial optimization because they are one of the first graph classes where it has been proved (independently by Sbihi [39] and Minty [31]) that a stable set of maximum cardinality can be found in polynomial time, using a combinatorial algorithm.

The structure of claw-free graphs has been extensively studied in a series of papers by Chudnovsky and Seymour (see [8] for a survey). Their result involves a lot of graph classes that are not of interest in this thesis, thus we postpone a more accurate analysis of this result to Chapter 4.

A relevant subclass of claw-free graphs is the class of *quasi-line* graphs. A graph is quasi-line if for every $v \in V$, $N[v]$ can be covered with two cliques. Trivially we cannot have a claw in a quasi-line graph, but some claw-free graphs are not quasi-line (we can see an example in Figure 1.2)

Quasi-line graphs have been also studied in terms of graph structure (again by Chudnovsky and Seymour [10]) and in terms of finding good upper

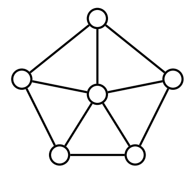


Figure 1.2: A claw-free graph which is not quasi-line

bounds for the coloring number $\chi(G)$ ([24, 5]). Quasi-line graphs are also a superclass of the very well known class of line graphs. A graph $L(G)$ is a line graph if it can be obtained as the intersection graph of the edges of a non necessarily simple graph G (G instead is called the *root graph* of $L(G)$). Again trivially any line graph is quasi-line but there are some quasi-line graphs that are not line (see Figure 1.3 for an example)

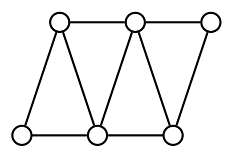


Figure 1.3: A quasi-line graph which is not line

The most relevant result on quasi-line graphs for this thesis is an algorithmic decomposition theorem presented in [17]. The result in [17] says that a graph G quasi-line either is net-free (a *net* is pictured in Figure 1.4) or it admits a strip decomposition, and this strip decomposition can be found in polynomial time (for a definition of strip decomposition see Chapter 3). We will go on further details on this result in Chapter 4.

1.3 Perfect Graphs

The clique number $\omega(G)$ and the coloring number $\chi(G)$ of a graph $G(V, E)$ are related by the inequality $\omega(G) \leq \chi(G)$, because in order to color all the vertices of the graph we need at least to assign a different color to each vertex of a clique of maximum size. Strict inequality can occur, for instance,

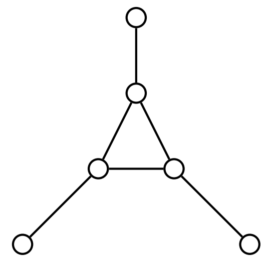


Figure 1.4: A net

for any odd cycle of length at least five, and its complement. We can instead always produce a graph where equality occurs, by adding to a graph G a clique of size $\chi(G)$, disjoint from V.

However the case when equality occurs becomes much more interesting and powerful when we require that equality holds also for all the induced subgraphs of G. Berge [4] defined a graph $G(V, E)$ to be *perfect* if $\omega(G') = \chi(G')$ holds for every induced subgraph G' of G.

Various classes of graphs could be shown to be perfect, among those the class of bipartite graphs and line graphs of bipartite graphs. Berge in [3, 4] observed that for all those classes, also the complementary graphs are perfect, thus conjectured what it is now known as the *perfect graph theorem*:

Theorem 1.1. *[29] A graph is perfect if and only if its complement is perfect.*

Theorem 1.1 has been proved by Lovász in [29]. As we have already seen, from the definition of perfect graphs it was straighforward from existing theorems that some important classes of graphs were perfect (e.g. bipartite graphs, complements of bipartite graphs, line graphs of bipartite graphs). It was also clear that some graphs where not perfect, for example odd holes and odd antiholes. Berge and P.C. Gilmore in [4] conjectured that a graph is perfect if and only if it is odd holes and odd antiholes free. Nowadays odd holes and odd antiholes free graphs are called *Berge*. Necessity of the conjecture is trivial, but sufficiency is far from being trivial. This conjecture, named *strong perfect graphs conjecture* in fact has been open for almost fourty years, and it has been proved in a huge piece of work by Chudnovsky, Robertson, Seymour and Thomas [7].

Theorem 1.2. *[7] A graph is perfect if and only if is Berge.*

The previous result, which gives also a characterization of perfect graphs in terms of *minimally imperfect* graphs (a graph G is minimally imperfect if G is not perfect but all its induced subgraphs are perfect), will be extensively used in this thesis.

1.4 Matchings

Given a graph $G(V, E)$, a subset M of E is called a *matching* if any two edges in M are disjoint. An important concept in finding a matching of maximum cardinality (i.e. a maximum matching) is that of an augmenting path. We say that a vertex $u \in V$ is covered by a matching M if there exists an edge uv for some $v \in V$ such that $uv \in M$.

Let M be a matching in a graph $G(V, E)$. A path P in G is called *M-augmenting* if P has odd length, its ends are not covered by M, and its edges are alternatingly out of and in M.

The relevance of augmenting paths is due to the following theorem of Petersen [37].

Theorem 1.3. *Let $G(V, E)$ be a graph and let M be a matching in G. Then either M is a matching of maximum size or there exists an M-augmenting path.*

This theorem has a straightforward algorithmic consequence: if we have an algorithm that either finds an M-augmenting path for any matching M or decide that it does not exists, then we can find a maximum size matching. Finding an M-augmenting path can be done in polynomial time in general graphs, thanks to Edmonds' algorithm [13], which solves the problem in $O(|V|^2|E|)$ (for a $O(|V|^3)$ implementation of the same algorithm see [41]).

Suppose we are also given a weight function $w : E \rightarrow \mathbb{R}^+$ on the edges of G, then one may ask if we can efficiently find a matching of G such that $w(M) = \sum_{e \in M} w(e)$ is maximum. The answer to this question is affirmative and was given again by Edmonds [14]. The algorithm for the maximum weight matching is a primal-dual algorithm and it runs in the original version in time $O(|V|^2|E|)$ (again for a $O(|V|^3)$ implementation of the same algorithm see [41]).

1.5 Stable sets

In a graph $G(V, E)$ a *stable set* is a set of vertices any two of which are non adjacent. The maximum size of a stable set in G is called the *stable set number* or *stability number* of G, and is denoted by $\alpha(G)$. Determining the stable set number is NP-complete (it can be shown via a reduction from a satisfiability problem, see [41]). Nevertheless there some classes of graphs where the problem is polynomially solvable and among those we have the class of claw-free graphs and the class of perfect graphs.

In the following we will analyze some special features of stable sets in claw-free graphs. Given a stable set S in a claw-free graph $G(V, E)$, for every vertex $v \in V \setminus S$, $|N(v) \cap S| \leq 2$ holds. We say that a vertex v is *superfree* if $N(v) \cap S = \emptyset$, is *free* if $|N(v) \cap S| = 1$ and it is *bound* if $|N(v) \cap S| = 2$. The property that in claw-free graphs any vertex has at most two neighbors in any stable set is relevant also from an algorithmic point of view.

Let $G(V, E)$ be a claw-free graph and let S be a stable set in G. A walk $P = v_0, v_1, \ldots, v_k$ (given by its vertex-sequence) is called *S-alternating* if precisely one of v_{i-1}, v_i belongs to S, for each $i = 1, \ldots, k$. It is an *S-augmenting path* if moreover P is a path, $v_0, v_k \notin S$ and $(S \setminus \{v_1, v_3, \ldots, v_{k-1}\}) \cup \{v_0, v_2, \ldots, v_k\}$ is stable. This implies that (if $k \geq 2$) each of v_0 and v_k has precisely one neighbor in S, and each of $v_2, v_4, \ldots, v_{k-2}$ precisely two. We can similarly define S-alternating cycles. S-augmenting paths are relevant because of the following result (for a proof see [41]).

Lemma 1.4. *If G is a claw-free graph with a stable set S, then there is a stable set larger than S if and only if there exists an S-augmenting path.*

The notion of augmenting paths somehow links the maximum stable set problem in claw-free graphs to the maximum matching in general graphs. If we consider a line graph G and we want to find a maximum stable set of G, then we can build the root graph of G and find a maximum matching of it.

Thus, as line graphs are a subclass of claw-free graphs, finding a maximum stable set in a claw-free graph is a generalization of the problem of finding a maximum matching.

If in addition to the graph $G(V, E)$ we are also given a weight function on the vertices $w : V \to \mathbb{R}^+$, then one may be interested on which is the stable set S of G with $w(S)$ maximum, that is a *maximum weighted stable set*. We denote with $\alpha_w(G)$ the weight of a maximum weighted stable set, and again determining $\alpha_w(G)$ is NP-complete in general graphs but the problem is polynomially solvable in claw-free graphs and in perfect graphs.

It is a relevant and known fact that in perfect graphs maximum weighted stable sets (and consequently also a maximum stable set) are precisely the optimal solutions of the following linear program:

$$\max \sum_{v \in V} w(v) x_v$$
$$\sum_{v \in C} x_v \leq 1 \quad \forall C \in \mathcal{K}(G)$$
$$x_v \geq 0 \quad \forall v \in V$$

Where $\mathcal{K}(G)$ is the family of all the maximal cliques of G. This linear program has an exponential number of constraints, moreover in perfect graphs the separation problem on this formulation reduces again to a maximum weighted stable set, so the naïve approach of applying the ellipsoid method to this program does not work. Nevertheless the problem is polynomially solvable using Lovász $\theta(G)$ function.

Another property of stable sets in perfect graphs of particular interest for this thesis is the following:

Property 1. *A graph G is perfect if and only if for each induced subgraph H of G there exists a stable set S_H such that $\omega(H[V(H) \setminus S_H]) < \omega(H)$.*

If we consider the complement graph $\overline{G}$, which is perfect again, the last property translates as follows:

Property 2. *A graph G is perfect if and only if for each induced subgraph H of G there exists a clique K_H such that $\alpha(H[V(H) \setminus K_H]) < \alpha(H)$.*

Property 2 states that in a perfect graph G we always have a clique that intersects all the maximum stable sets of G. We will call such a clique *crucial*.

1.6 Clique covers

A clique cover $(\mathcal{K}, y)$ of G is a collection $\mathcal{K}$ of cliques, with a non-negative weight y_C assigned to each clique $C \in \mathcal{K}$, such that, for each vertex v of G, $\sum_{C \in \mathcal{K}: v \in C} y_C \geq 1$. A clique cover is minimum if the sum of all the weights assigned to the cliques in $\mathcal{K}$ is minimum. We will denote with $\tau(G)$ the sum of all the weights assigned to the cliques in $\mathcal{K}$ in a minimum clique cover $(\mathcal{K}, y)$. Determining $\tau(G)$ is NP-complete in general graphs. We observe that if y_C is integer for every $C \in \mathcal{K}$, then the set of cliques with $y_C > 0$ is a set of cliques covering V, in the sense that $\bigcup_{C \in \mathcal{K}: y_C > 0} C \supseteq V$.

If we are also given a weight function on the vertices $w : V \to \mathbb{R}^+$, a *weighted clique cover* is a collection of cliques $\mathcal{C}$ of G, each with an associated value y_C, such that $\sum_{C \in \mathcal{C}: v \in C} y_C \geq w(v)$ for every $v \in V$. We say that a weighted clique cover is minimum if its *value* $\sum_{C \in \mathcal{C}} y_C$ is minimum. We denote with $\tau_w(G)$ the value of a minimum weighted clique cover of G; again determining $\tau_w(G)$ is NP-complete in general graphs.

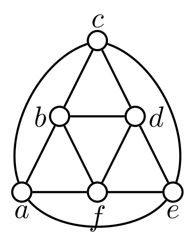
In general graphs $\alpha(G) \leq \tau(G)$, because two vertices of a stable set must be covered by different cliques and similarly $\alpha_w(G) \leq \tau_w(G)$. In perfect graphs $\alpha(G) = \tau(G)$ and $\alpha_w(G) = \tau_w(G)$, because the minimum weighted clique cover problem has the following linear programming formulation (where again $\mathcal{K}(G)$ is the family of all the maximal cliques of G):

$$\begin{aligned} \min \sum_{C \in \mathcal{K}(G)} y_C \\ \sum_{C \in \mathcal{K}(G): v \in C} y_C \geq 1 \quad \forall v \in V \\ y_C \geq 0 \quad \forall C \in \mathcal{K}(G) \end{aligned}$$

which happens to be exactly the dual of the linear program of the maximum weighted stable set in perfect graphs. It follows that we can determine $\tau_w(G)$ in perfect graphs using Lovász $\theta_w(G)$ function in polynomial time.

If we want to compute also a minimum weighted clique cover of a perfect graph G (and not only the number $\tau_w(G)$), we can use a polynomial algorithm proposed by Groetschel, Lovász and Schrijver in [21]. This algorithm is not combinatorial and it uses the $\theta_w(G)$ function combined with other techniques.

Another important property of the linear programming formulation for the minimum weighted clique cover is that, if the graph G is perfect, then for every *integer* weight function $w : V \to \mathbb{N}$ there exists an *integer* optimal solution. This property follows from a result of Fulkerson (see [18]) on antiblocking pairs of polyhedra. We underline that the property we have just mentioned does not mean that the polyhedron over which we are optimizing is integral. In fact it is easy to see that it may have some fractional vertices. Consider the following graph



and the integer weight function that assigns a weight of 1 to every vertex. Let us call $K_1 = \{a, b, c\}$, $K_2 = \{c, d, e\}$, $K_3 = \{a, e, f\}$ and $K_4 = \{b, d, f\}$. Consider the feasible point $y_{K_1} = y_{K_2} = y_{K_3} = y_{K_4} = \frac{1}{2}$: this point is a fractional vertex of our polyhedron, because the only integer vertex that has $y_{K_4} \neq 0$ has also $y_{\{a,c,e\}} \neq 0$, and thus the point $y_{K_1} = y_{K_2} = y_{K_3} = y_{K_4} = \frac{1}{2}$ cannot be obtained as a convex combination of other vertices.

From duality between the maximum weighted stable set problem and the minimum weighted clique cover problem we can also derive an interesting property of the latter. Let us consider a maximum weighted stable set S and a minimum weighted clique cover: then every clique K in G with $y_K > 0$ must intersect S, otherwise we would violate the complementary slackness condition $y_K \cdot (x(K) - 1) = 0$. It follows that in perfect graphs we can always express the family of cliques $\mathcal{K}$ with $y_K > 0$ as the union over $s \in S$

of subfamilies $\mathcal{K}_s$, where $\mathcal{K}_s = \{K \in \mathcal{K} : s \in K\}$.

1.7 Claw-free perfect graphs

The class of claw-free perfect graphs was studied extensively in the past. The first structural result for this class was obtained by Chvatal and Sbihi in [12], where they proved that every claw-free Berge graph can be decomposed via clique-cutsets into two types of graphs: *elementary* and *peculiar* (as we will not deal with those graph classes we skip the definition of them). In a successive paper by Maffray and Reed [30] the structure of elementary graphs is investigated and as a consequence an alternative proof that claw-free Berge graphs are perfect is given (the first proof was due to Parthasarathy and Ravindra [36]). Still, with those results, it was not possible to set a structure theorem for claw-free perfect graphs, because composing elementary and peculiar graphs via clique-cutsets can lead to a claw. Nevertheless a structure theorem for claw-free perfect graphs was finally settled by Chudnovsky and Plumettaz in [6].

It is remarkable that Hsu and Nemhauser in [22] were able to give another proof that claw-free Berge graphs are perfect showing directly that for a claw-free Berge graph G $\omega(G') = \chi(G')$ holds for every induced subgraph G' of G.

One of the properties of claw-free perfect graphs, proved by Hsu and Nemhauser in the same paper is the following:

Lemma 1.5. *If G is a claw-free perfect graph, then G is quasi-line.*

Proof. Let us consider a vertex v: as G is claw-free $\alpha(G[N(v)]) \leq 2$, but then as G is perfect, if $\alpha(G[N(v)]) = 1$ $G[N[v]]$ is a clique, and if $\alpha(G[N(v)]) = 2$ $G[N[v]]$ is a cobipartite graph. It follows that for every $v \in V$, $N[v]$ can be covered by two cliques. $\square$

We will see in Chapter 2 that this lemma is of particular interest when one wants to compute a minimum clique cover of a claw-free perfect graph. We underline that while for quasi-line graphs there exists an algorithmic decomposition theorem, this is not the case for claw-free perfect graphs, because the decomposition of Chudnovsky and Seymour is not algorithmic. Moreover we could not find, to the best of our efforts, a way to exploit the characterization of the subclasses of claw-free perfect graphs for the computation of the minimum weighted clique cover of claw-free perfect graphs. This is why, when we refer to claw-free perfect graphs, we often refer to more general theorems and decompositions for quasi-line (perfect and non-perfect) graphs.

Chapter 2

The MCC problem on claw-free perfect graphs

2.1 Introduction

Given a graph G, a *clique cover* of G is a collection of cliques, with a non-negative weight y_C assigned to each clique C in the collection, such that, for each vertex v of G, the sum of the weights of the cliques containing v in the collection is at least one. A *minimum clique cover* of G (MCC) is a clique cover such that its *value* (the sum of the weights of all the cliques in the collection) is minimum. If all the weights assigned to the cliques are integer (i.e. zero or one), then a minimum clique cover can be interpreted as a collection of minimum cardinality of cliques, such that each vertex of the graph is contained in at least one clique in the collection.

The MCC problem on perfect graphs is the dual of the maximum stable set problem (MSS for short), thus for every perfect graph G the value of the MCC, that we indicate with $\tau(G)$, is equal to the stability number $\alpha(G)$. Moreover it can be shown that there always exists a minimum clique cover that assigns an integer weight to the cliques of the graph.

The crucial property that is often used to tackle the problem on perfect graphs is the following: let $\mathcal{K}(G)$ be the collection of all the maximal cliques of G and let y be a MCC of G. For every clique $C \in \mathcal{K}(G)$ with $y_C = 1$ and every MSS S of G $C \cap S \neq \emptyset$.

Definition 2.1. *Let C be a clique of G. If $C \cap S \neq \emptyset$ for every maximum stable set S of G, then C is a* crucial *clique.*

We can deduce that in an integer MCC (that is a MCC where $y_C \in \{0, 1\}$ for every $C \in \mathcal{C}$), every clique with $y_C = 1$ is a crucial clique, and viceversa every vertex $v \in S$ for some MSS S of G must be contained in some crucial clique.

In fact a general technique to solve the MCC problem in perfect graphs could be to iteratively find a crucial clique and delete it. By duality we know that we would repeat this step at most $\alpha(G) \leq |V|$ times. In this framework arises the problem to find a combinatorial algorithm for claw-free perfect graphs; it is known that in claw-free graphs the MSS problem is polynomially solvable via a combinatorial algorithm. The first combinatorial algorithm for the MCC in claw-free perfect graphs has been proposed by Hsu and Nemhauser in [22] and exploits the notion of crucial cliques and Lemma 1.5.

The algorithm in [22] starts with a MSS of G and then computes a MCC of G by iteratively finding and deleting crucial cliques. In particular, fixed a vertex $s \in S$, they want to find a crucial clique containing s, and this clique must be contained in $N[s]$. The algorithm they develop is based on the fact that in claw-free graphs we can answer to the following question in polynomial time: let $v \in V$, is v contained in some MSS of G? In particular they suggest to answer to this question looking for augmenting paths that have an extreme in v (which was the most efficient way to look at this problem at those times). It is nevertheless interesting to notice that in every subclass of perfect graphs where it is easy to determine if a vertes or a subset of vertices belong to a MSS, a similar approach can be used and maybe more efficient techniques can be applied to the basic step of finding a crucial clique in $N[s]$, for some $s \in S$ where S is a MSS of G. The algorithm of Hsu and Nemhauser runs in $O(|V|^5)$, where V is the set of vertices of G (in [22] is actually claimed that the running time is $O(|V|^{5.5})$, but a more accurate analysis of Minty's algorithm (see [41]) shows that the actual running time is $O(|V|^5)$).

In this chapter we will make use of directed graphs. We give here some basic definitions, for all the non mentioned definitions about directed graphs the reader may refer to the corresponding definitions for undirected graphs or to [41]. A *directed graph* is an ordered pair $D(\mathcal{N}, A)$ where $\mathcal{N}$ is a set of *nodes* and A is a set of ordered pairs of vertices each of which is called *arc* (that is the arc uv is different from the arc vu). Given a directed graph $D(\mathcal{N}, A)$ and an integer $k \in \mathbb{N}$, an ordered set of nodes $v_1, \ldots, v_k$ is a *directed walk* (*of length* $k-1$) if $(v_i, v_{i+1}) \in A(D)$ for each $i \in [k-1]$. If $v_1, \ldots, v_k$ are all distinct, the walk is called a *path*. We say that a path $P = v_1, \ldots, v_k$ is *chordless* if moreover there are no arcs (v_i, v_j), for every i, j such that $j - i \geq 2$. A directed graph is *strongly connected* if for every pair of nodes $\{u, v\}$ there exists a directed walk from u to v and a directed walk from v to u. A maximal strongly connected subgraph of D is a *strongly connected component* of D.

In next section we give a short introduction to the 2-SAT problem and then, in Section 2.3 we introduce a new algorithm for the MCC problem on claw-free perfect graphs. The aim of this algorithm is to grow at the same time a MCC and a MSS of G and it runs in $O(|V|^3)$, thus improving the complexity bound for the problem.

2.2 A short introduction to the 2-SAT problem

Given a set of boolean variables $U = \{u_1, \ldots, u_p\}$, we say that a variable $u_i \in U$ or its negation, that we denote with $\neg u_i$, are *terms* and a disjunction (that we indicate with $\vee$) of a subset of terms of U is a *clause*. We call a conjunction (that we indicate with $\wedge$) of clauses a *formula*; in particular a formula, as we have defined it, is a conjuction of disjunction of terms. We can always assume that formulas are conjuctions of disjunction of terms, that is we can always assume that they are in the *conjuctive normal form* (CNF for short). Finally a truth assignment is a function $t : U \rightarrow \{true, false\}$, that assigns to each variable in U a value *true* or *false*.

A *satisfiability* problem looks for an answer to the following question: given a formula, is there any truth assignment to the variables such that the formula is true? The satisfiability problem is NP-hard (see [20]) even when each clause contains only three terms. The *2-satisfiability* (or 2-SAT) problem is a satisfiability problem where each clause has exactly two terms:

it is polynomially solvable, in a time which is linear in the number of clauses by an algorithm of Aspvall, Plass and Tarjan [1]. This technique is based on the construction of the (directed) *implication graph* D of the given instance. The implication graph D has the following structure: for every variable and every negation of a variable there is a node, and for each clause $w_i \vee w_j$ there are the arcs $\neg w_i w_j$ and $\neg w_j w_i$. After building the implication graph D it is necessary to find the strongly connected components of it. The key observation is that every strongly connected component corresponds to a set of terms that must have the same value. Then one can build the condensation of the implication graph, find a topological order of it and assign values according to this topological order to the strongly connected components of the implication graph in the following way: set the variables in a strongly connected component W to *true* if W appears after the corresponding negated component $\bar{W}$ in the topological order, and to $false$ otherwise. With this procedure one can obtain a truth assignment of the 2-SAT instance (if any).

2.3 An algorithm for the MCC in quasi-line perfect graphs via 2-SAT

We are given a stable set S of a claw-free perfect graph $G(V, E)$, and without loss of generality we assume that S is maximal and there are no augmenting paths of length 2. For a vertex $v \in V \setminus S$ that is bound, we let $s_1(v)$ and $s_2(v)$ be its neighbors in S; for a vertex $v \in V \setminus S$ that is free, we let $s(v)$ be its neighbor in S.

Our target is the following. We want to check if S is a maximum stable set of G. In case it is, we want to build a suitable clique cover of G of size $|S|$; in case it is not, we want to find an augmenting path.

We will achieve our target by formulating a suitable instance of the 2-SAT problem. We will express the formula for the 2-SAT problem in the 2-conjunctive normal form, that is every clause is composed by two terms and the formula is a conjunction of disjunctions. For every bound (resp. free) vertex $v \in V \setminus S$, we define two variables, or terms, $x_{vs_1(v)}$ and $x_{vs_2(v)}$ (resp. $x_{vs(v)}$) and we say that e.g. $x_{vs_1(v)}$ is *true* if and only if v is covered with a clique containing $s_1(v)$. We denote by $\neg x_{vs}$ the negation of a term x_{vs}. We also introduce an "auxiliary" variable y. We consider three classes of clauses:

(c1) for each $u \in V \setminus S$ that is bound, $x_{us_1(u)} \vee x_{us_2(u)}$ must be true;

(c2) for each $s \in S$ and each $u, v \in N(s)$ that are non-adjacent, $\neg x_{us} \vee \neg x_{vs}$ must be true;

(c3) for each $u \in V \setminus S$ that is free, both $x_{us(u)} \vee y$ and $x_{us(u)} \vee \neg y$ must be true (i.e., $x_{us(u)}$ must be true).

Consider the 2-SAT instance made of the conjunction of all the above clauses, which we denote in the following by the pair (G, S). It is straightforward to check that a clique cover of size $|S|$ (in case it exists, it is a minimum clique cover of G), induces a solution (i.e. a satisfying truth assignment) to (G, S). Vice versa, from a solution to (G, S) we can easily build a clique cover of size $|S|$ of G. In fact, for each vertex $s \in S$, let $X(s) := \{s\} \cup \{v \in N(s) : x_{vs} \text{ true}\}$. Note that for each free vertex u, following the clauses $(c3)$, $u \in X(s(u))$. Moreover, for each $s \in S$, $X(s)$ is a clique, following the clauses $(c2)$. Finally, following the clauses $(c1)$, each bound vertex u belongs to either $X(s_1(u))$ or to $X(s_2(u))$. The family $\{X(s), s \in S\}$ is then a clique cover of size $|S|$, i.e. a minimum clique cover of G.

Therefore a maximal stable set S is a maximum stable set of G if and only if there exists a solution to the 2-SAT instance (G, S). Moreover, from a solution to (G, S) we can easily build that minimum clique cover of G.

Following the above discussion, in order to design an algorithm for the minimum clique cover problem of a claw-free perfect graph G, we are left with the following question: what if S is *not* a maximum stable set of G, i.e. there is no solution to the 2-SAT instance (G, S)? As we show in the following, in this case, the implication graph of (G, S) "suggests" a path of G that is augmenting with respect to S. We denote by D be the implication graph of (G, S) and let $\overline{D} = D[V(D) \setminus \{y, \neg y\}]$. The implication graph D has a very particular structure and every arc of D belongs to one of the following classes:

1. Arcs that go from a *positive* term (that is a non negated variable) x_{vs} with $v \in V \setminus S$ and $s \in S \cap N(v)$ to a *negative* term (that is the negation of a variable) $\neg x_{ws}$ with $w \in V \setminus S$ and $s \in S \cap N(w)$.

2. Arcs that go from a *negative* term $\neg x_{vs_1(v)}$ with $v \in V \setminus S$ bound to a *positive* term $x_{vs_2(v)}$.

3. Arcs that go from a *negative* term $\neg x_{vs(v)}$ with $v \in V \setminus S$ free to y and to $\neg y$.

4. Arcs that go from y and $\neg y$ to a *positive* term $x_{vs(v)}$ with $v \in V \setminus S$ free.

Lemma 2.2. *Let u and v two free vertices of G. An augmenting path in G between u and v corresponds in $\overline{D}$ to a chordless directed path from $x_{us(u)}$ to $\neg x_{vs(v)}$. Vice versa, a chordless directed path in $\overline{D}$ from $x_{us(u)}$ to $\neg x_{vs(v)}$ corresponds in G to augmenting path between u and v.*

Proof. The first statement is trivial. Now let P be a directed path of $\overline{D}$ from $x_{us(u)}$ to $\neg x_{vs(v)}$, with u and v free. By construction, $P = x_{u_0s_0}, \neg x_{u_1s_0}, x_{u_1s_1}, \neg x_{u_2s_1}, ..., x_{u_ks_k}, \neg x_{u_{k+1}s_k}$, with $u_0 \equiv u$ and $u_{k+1} \equiv v$ free, $s_0 = s(u_0), s_k = s(v)$ and $\{s_1(u_i), s_2(u_i)\} = \{s_{i-1}, s_i\}$, for $i = 1, \ldots, k$. Therefore, P induces on G a walk $Q = u_0, s_0, u_1, s_1, \ldots, u_k, s_k, u_{k+1}$.

Claim 2.3. *If the vertices of Q are different from each other, then Q is an augmenting path.*

Proof. Suppose the contrary. There must be either an edge u_is_j, with $i \neq j, j+1$, or an edge u_iu_h with $|i-h| \geq 2$. In the former case, either there is a claw, or u (v) is bound, a contradiction. So, assume there are no edges u_is_j, with $i \neq j, j+1$. By construction, we know also that there is no edge u_iu_{i+1}. Suppose there is an edge u_iu_h with $|i-h| \geq 2$ and choose such an edge as to minimize $|i-h|$. Then there is an odd hole in G, a contradiction. □

We will indeed show that the vertices of Q are different from each other.

Claim 2.4. *There is no g, $0 \leq g \leq k$, such that both $x_{u_gs_g}$ and $\neg x_{u_gs_g}$ belong to P.*

Proof. For, suppose to the contrary. W.l.o.g let $P' = x_{u_0 s_0}, \dots, x_{u_g s_g}, ...,$ $x_{u_h s_h}, \neg x_{u_{h+1} s_h} \equiv \neg x_{u_g, s_g}$. Note that the path P' is a directed path also for the implication graph $D' = D(G', S')$ of the 2-SAT "sub-instance" associated with the pair $(G' = G[\{u_0, s_0, ..., u_h, s_h\}], S' = S \cap \{s_0, ..., s_h\})$. Such a path shows that S' is not a maximum stable set for G'. (Otherwise, G' would have a clique cover of size $|S'|$, which has to "assign" u_0 to s_0. But this leads to both $x_{u_g s_g}$ and $\neg x_{u_g s_g}$ being true, which is impossible.) But that is a contradiction, as every vertex of $V(G') \setminus S'$ but u_0 is bound w.r.t. S', and so in G' there are no augmenting paths w.r.t. S'. □

Claim 2.5. *The vertices $u_0, u_1, \dots, u_k, u_{k+1}$ are different from each other.*

Proof. Suppose now by contradiction that there are l and j with $u_l = u_j$ and $k+1 \geq l > j \geq 0$. First suppose that both u_l and u_j are bound, therefore $k+1 > l > j > 0$, and the nodes $\neg x_{u_j s_{j-1}}$ and $\neg x_{u_l s_{l-1}}$ belong to P. Note that either $s_{l-1} = s_{j-1}$ or $s_{l-1} = s_j$. In the former case, P is not simple, as it visits twice the node $\neg x_{u_j s_{j-1}}$. In the latter case, both the node $x_{u_j s_j}$ and the node $\neg x_{u_j s_j}$ belong to P, a contradiction to Claim 2.4. The case where $u_0 = u_{k+1}$ leads to a contradiction along the same lines as the latter case. □

So, in order to prove the lemma, we are left with showing that the vertices of $Q \cap S$ are different from each other. The following claim directly follows from the hypothesis that P is chordless.

Claim 2.6. *If there is a vertex $s_i = s_h$ with $1 \leq i < h \leq k$, then $u_i u_{h+1} \in E$.*

We now suppose by contradiction that there is at least one pair $\{j, l\}$, with $1 \leq j < l \leq k$ and $l - j \geq 2$, such that $s_j = s_l$. We choose the pair $\{j, l\}$ as to minimize $l - j$, and we break ties in favour of the pair with j smaller.

Claim 2.7. *$C = s_j, u_{j+1}, s_{j+1}, \dots, s_{l-1}, u_l, s_j$ is an alternating chordless cycle.*

Proof. The vertices of C are distinct, following our choice of j and l. Therefore, C is an alternating cycle. Also it has no chords: this is trivial if $l - j = 2$; otherwise, it follows either from claw-freeness or from perfection (in particular, if a vertex $u_i \in V(C)$ is adjacent to a vertex $u_h \in V(C)$, with $|h - i| \geq 2$, then there would be either a claw or an odd hole). □

In the following we often rely on the next claim, whose simple proof (based on claw-freeness) we skip.

Claim 2.8. *Every vertex of $V \setminus C$ is either adjacent to exactly two vertices of C that are consecutive, or to exactly three vertices of C that are consecutive, or to four vertices of C and they are either consecutive or made of two pairs of consecutive vertices.*

Claim 2.9. $j = 0$*, i.e.* $s_l = s_0$*. Moreover,* $N(u_0) \cap V(C) = \{s_0, u_l\}$ *and* $N(u_{l+1}) \cap V(C) = \{s_0, u_1, s_1\}$*. In particular,* $s_{l+1} = s_1$*.*

Proof. The vertices u_j, u_l, u_{j+1} and u_{l+1} belong to $N(s_j)$. Note that, by construction, $u_j u_{l+1} \in E$, while $u_j u_{j+1}, u_l u_{l+1} \notin E$. Also, $u_{j+1} u_l \notin E$, as it would be a chord for C. So, in order to prevent the claw $(s_j; u_l, u_{l+1}, u_{j+1})$ (resp. $(s_j; u_l, u_j, u_{j+1})$), we must have that $u_{l+1} u_{j+1} \in E$ (resp. $u_j u_l \in E$).

Suppose that u_j is bound, i.e. $j > 0$. Note that our choice of j and l is such that $s_{j-1} \notin V(C)$. Then, in order to avoid the claw $(u_j; u_l, s_{j-1}, u_{l+1})$, we must have that $u_{l+1} s_{j-1} \in E$, and so $s_{l+1} = s_{j-1}$. Now observe that it follows from Claim 2.8 that $N(u_j) \cap V(C) = \{s_j, u_l\}$ and $N(u_{l+1}) \cap V(C) = \{s_j, u_{j+1}\}$. But then $u_j, u_{l+1}, u_{j+1}, s_{j+1}, \ldots, s_{l-1}, u_l, u_j$ is an odd hole, a contradiction.

Therefore, $j = 0$, and $s_l = s_0$. It follows from Claim 2.8 that u_0 is adjacent only to s_0 and u_l. Note that $u_0, s_0, u_1 \in N(u_{l+1})$. We claim that u_{l+1} is bound. If not, from Claim 2.8, we have that $N(u_{l+1}) \cap V(C) = \{s_0, u_1\}$: but then $u_0, u_{l+1}, u_1, s_1, \ldots, s_{l-1}, u_l, u_0$ is an odd hole, a contradiction. Therefore u_{l+1} is bound. Since $(u_{l+1}; u_0, u_1, s_{l+1})$ is not a claw, $s_{l+1} u_1 \in E$, i.e., $s_{l+1} = s_1$, and $N(u_{l+1}) \cap V(C) = \{s_0, u_1, s_1\}$. □

Claim 2.10. *$N(u_{l+2}) \cap V(C) = \{u_1, s_1 \equiv s_{l+1}, u_2, s_2 \equiv s_{l+2}\}$. In particular, u_{l+2} is bound.*

Proof. By construction and Claim 2.9, $u_{l+2} \in N(s_1)$ and, from Claim 2.6, $u_1 u_{l+2} \in E$. From Claim 2.9, we have that $u_{l+1} u_2 \notin E$. Therefore, to avoid the claw $(s_1; u_2, u_{l+1}, u_{l+2})$, we have that $u_2 u_{l+2} \in E$. We claim that u_{l+2} is bound. If not, from Claim 2.8, we have that $N(u_{l+2}) \cap V(C) = \{u_1, s_1, u_2\}$: but then $u_0, u_{l+1}, u_1, u_{l+2}, u_2, s_2, \dots, s_{l-1}, u_l, u_0$ is an odd hole, a contradiction.

Therefore u_{l+2} is bound and claw-freeness shows that either (i) $s_{l+2} \equiv s_2$ (note that, if $l = 2$, then $s_{l+2} \equiv s_0$) or (ii) $l > 2$ and $s_{l+2} \equiv s_0$. If (i) holds, we are done because we have shown that $N(u_{l+2}) \cap V(C) \subseteq \{u_1, s_1, u_2, s_2\}$, but from Claim 2.8, a vertex cannot be adjacent to more then four consecutive vertices in C. If (ii) holds, then it follows from claw-freeness that $u_l u_{l+2} \in E$. But then u_{l+2} is adjacent to three non consecutive vertices in C, u_1, u_2, u_l, contradicting Claim 2.8. □

Now we must delve into two cases: $l > 2$ and $l = 2$. We first get rid of the case $l > 2$.

Claim 2.11. *Suppose that $l > 2$. For each $i = 2, 3, \dots, l-1$, $N(u_{l+i}) \cap V(C) = \{u_{i-1}, s_{i-1} \equiv s_{l+i-1}, u_i, s_i \equiv s_{l+i}\}$ and $\{u_{l+1}, u_{l+2}, \dots, u_{l+i}\}$ is a stable set.*

Proof. The proof is by induction. The case $i = 2$ follows from Claim 2.10. Now suppose that the statement holds for $2, 3, \dots, i < l-2$: we will show that it holds also for $i + 1 \leq l - 1$ (note that $l \geq 4$ else we are done). By construction, $u_{l+i+1} \in N(s_i)$, and, from Claim 2.6, $u_i u_{l+i+1} \in E$. Then, from claw-freeness, $u_{i+1} u_{l+i+1} \in E$. Therefore, $\{u_i, s_i, u_{i+1}\} \subseteq N(u_{l+i+1}) \cap V(C)$ and, from Claim 2.8, $N(u_{l+i+1}) \cap V(C) \subset \{s_{i-1}, u_i, s_i, u_{i+1}, s_{i+1}\}$. In any case, by claw-freeness and inductive hypothesis, it follows that u_{l+i+1} is non-adjacent to $\{u_{l+1}, u_{l+2}, \dots, u_{l+i}\}$; hence $\{u_{l+1}, u_{l+2}, \dots, u_{l+i+1}\}$ is a stable set.

We now claim that $N(u_{l+i+1}) \cap V(C) = \{u_i, s_i, u_{i+1}, s_{i+1}\}$. If not, then either u_{l+i+1} is free or u_{l+i+1} is adjacent to $s_{i-1} \equiv s_{l+i+1}$. In the former case, there is an odd hole with vertices: $u_0, u_{l+1}, u_1, u_{l+2}, \ldots, u_i, u_{l+i+1}, u_{i+1}, s_{i+1}, \ldots, u_{l-1}, s_{l-1}, u_l, u_0$. In the latter case, Claim 2.6 shows that $u_{i-1}u_{l+i+1} \in E$, a contradiction to the fact that $N(u_{l+i+1}) \cap V(C) \subset \{s_{i-1}, u_i, s_i, u_{i+1}, s_{i+1}\}$. □

Claim 2.11 shows that, if $l > 2$, then $u_{l+2}, u_{l+3}, \ldots, u_{2l-1}$ are bound. By the same arguments, it is possible to show that $\{u_{l-1}, s_{l-1}, u_l\} \subseteq N(u_{2l}) \cap V(C)$, $N(u_{2l}) \cap V(C) \subset \{s_{l-2}, u_{l-1}, s_{l-1}, u_l, s_l \equiv s_0\}$ and $\{u_{l+1}, u_{l+2}, \ldots, u_{2l}\}$ is a stable set. However, if $u_0u_{2l} \notin E$ then $u_0, u_{l+1}, u_1, u_{l+2}, \ldots, u_{l-1}, u_{2l}, u_l, u_0$ is an odd hole. Therefore $u_0u_{2l} \in E$; but in this case, there is an odd hole with vertices: $u_0, u_{l+1}, s_1, u_2, s_2, \ldots, u_{l-2}, s_{l-2}, u_{l-1}, u_{2l}, u_0$.

We therefore assume in the following that $l = 2$.

Claim 2.12. *Suppose $l = 2$. For each u_{l+i} with $i \geq 2$ even, $N(u_{l+i}) \cap V(C) = \{u_1, s_1 \equiv s_{l+i-1}, u_2, s_0 \equiv s_{l+i}\}$ and u_{l+i} is complete to $\{u_0, u_1, u_2, \ldots, u_{l+i-2}\}$. Similarly, for each u_{l+i} with $i \geq 2$ odd, $N(u_{l+i}) \cap V(C) = \{u_2, s_0 \equiv s_{l+i-1}, u_1, s_1 \equiv s_{l+i}\}$ and u_{l+i} is complete to $\{u_0, u_1, u_2, \ldots, u_{l+i-2}\}$.*

Proof. The proof is by induction. The case $i = 2$ follows from Claim 2.10 and because u_{l+2} has to be adjacent to u_0, in order to avoid the odd hole with vertices u_0, u_1, u_2, u_3, u_4. Now suppose that the statement holds for $i \geq 2$: we will show that it holds also for $i + 1$.

Suppose first that i is odd, that is $l + i + 1$ is even and by construction $u_{l+i+1} \in N(s_1)$. From Claim 2.6 u_{l+i+1} is adjacent to every u_b with $b < l+i$ and b odd. Thus we are left to show that u_{l+i+1} is also adjacent to every u_c with $c < l + i$ and even. Suppose by contradiction that there exists $c < l + i$ and even such that $u_cu_{l+i+1} \notin E$, and choose the largest one. Then by claw-freeness $c \neq l + i - 1$, and for the inductive hypothesis there is an odd antihole with vertices $u_c, u_{c+1}, \ldots, u_{l+i}, u_{l+i+1}, u_c$. Finally we are left with showing that $s_0 \in N(u_{l+i+1})$: suppose by contradiction that $s_0u_{l+i+1} \notin E$, then $s_0, s_1, u_0, u_1, u_2, \ldots, u_{l+i}, u_{l+i+1}, s_0$ is an odd antihole.

The case where i is even goes along the same lines as above. □

It follows from Claim 2.12 that all the vertices in $\{u_1, \ldots, u_k, u_{k+1}\}$ are bound, when $l = 2$. But this is a contradiction to u_{k+1} free. Then all vertices $s_0, s_1, \ldots, s_k$ are different from each other. □

The following lemma shows that finding the strong component of D containing y is indeed sufficient to check whether the 2-SAT instance (G, S) is feasible.

Lemma 2.13. *Let $C(y)$ be the strong component of D containing y. If u is a free vertex of G, then $x_{us(u)}$ belongs to $C(y)$ if and only if u is the extreme of an augmenting path of G.*

Proof. Let u be a free vertex of G, such that in G there is an augmenting path between u and another free vertex v. Following Lemma 2.2, in $\overline{D}$ there is a directed path P from $x_{us(u)}$ to $\neg x_{vs(v)}$ (and a directed path from $x_{vs(v)}$ to $\neg x_{us(u)}$). Note that trivially P is also a path of D. Also the arcs $\neg x_{vs(v)}y$ and $yx_{us(u)}$ belong to $A(D)$, because of the clauses (c3). Then there is a directed cycle of D containing y and $x_{us(u)}$, so they are in the same strong component of D, that is, $C(y)$.

Suppose now that u is a free vertex of G and $x_{us(u)}$ belongs to $C(y)$. Therefore, there is a directed path from $x_{us(u)}$ to y. By construction, each arc entering into y is of the form $\neg x_{vs(v)}y$, with v a free vertex of G. So in D there is a directed path from $x_{us(u)}$ to $\neg x_{vs(v)}$. Note that this path is also a directed path of $\overline{D}$, unless $\neg y$ belongs to it; however, in this case, as again each arc entering into $\neg y$ is of the form $\neg x_{zs(z)}\neg y$, with z a free vertex of G, there is a directed path from $x_{us(u)}$ to $\neg x_{zs(z)}$ that avoids y and $\neg y$, i.e., a directed path of $\overline{D}$. In both cases, it follows from Lemma 2.2 that u is the extreme of an augmenting path of G. □

We may therefore state our algorithm for the maximum stable set and the minimum clique cover problem in claw-free perfect graphs.

Lemma 2.14. *Algorithm 1 is correct and terminates in $O(|V(G)|^3)$.*

Proof. Correctness follows from our previous results. We now deal with complexity issues. We first observe that we repeat steps 1, 2 and 3, that

Algorithm 1

Require: A graph $G(V, E)$ that is claw-free and perfect and a maximal stable S set of G such that there is no augmenting path with length 2 w.r.t. S.

Ensure: A MCC and a MSS for $G(V, E)$.

1: Let D be the implication graph of the 2-SAT instance (G, S).

2: Let $C(y)$ be the strongly connected component of D containing y.

3: **If** there is a free vertex u of G such that $x_{us(u)}$ belongs to $C(y)$, then, following Lemma 2.13, there exists an augmenting path P leaving from u. Following Lemma 2.2, P can be found through a BFS in $\overline{D}$. Let $S \leftarrow V(P)\Delta S$ and go back to step 1.

4: **Else**, following Lemma 2.13, S is a MSS of G. A solution to the 2-SAT instance (G, S) can be found by the second part of the algorithm of Aspvall Plass and Tarjan. Then a minimum clique cover of G can be built by setting $X(s) := \{s\} \cup \{v \in N(s) : x_{vs} \text{ true}\}$ for each $s \in S$.

5: **Return** $\{X(s), s \in S\}$ and S.

is we augment S, at most $|V(G)|$ times. Each of these steps takes at most $O(|V(D)| + |A(D)|)$, in particular we can determine the strongly connected components of D in time $O(|V(D)| + |A(D)|)$ thanks to the algorithm of Tarjan [42]. Step 4 is performed only once and also takes $O(|V(D)|+|A(D)|)$ (see [1]), so the algorithm terminates in $O(|V(G)|(|V(D)| + |A(D)|))$. We finally observe that by construction $|V(D)| = O(|V(G)|)$ and $|A(D)| = O(|V(G)|^2)$, thus the bound given by the lemma is correct. □

Chapter 3

The MWCC problem on strip-composed perfect graphs

In this chapter we present an algorithmic theorem for the minimum weighted clique cover (MWCC for short) problem on strip-composed perfect graphs. We show that, similarly to what has been done for the maximum weighted stable set (MWSS for short) in strip-composed graphs in [34], we can find a MWCC of a strip-composed perfect graph in polynomial time if we can solve in polynomial time the same problem on each strip and if the strip decomposition of G is given. Moreover if the weight function is integer we can find an integer MWCC.

In this chapter we make use also of multigraphs (the root graph of a simple line graph can be a multigraph). We recall that a *star* or a *multistar* is the set of edges incident to a vertex v, while a *triangle* or *multitriangle* is a complete graph on three vertices with eventually multiple edges.

3.1 Strip-composed graphs

Chudnovsky and Seymour in [8] introduced a strip composition operation in order to define their decomposition result for claw-free graphs. The operation of strip composition can be generalized also to non-claw-free graphs, and in the last years it has become a powerful mean to deeply understand the structure of some graph classes and to develop a new decomposition approach for some combinatorial problems (f.e. see [34], for the MWSS). First we borrow some definitions from the work of Chudnovsky and Seymour.

Definition 3.1. *A strip $H = (G, \mathcal{A})$ is a graph G (not necessarily connected) with a multi-family $\mathcal{A}$ of either one or two designated non-empty cliques of G. The cliques in $\mathcal{A}$ are called the* extremities *of H, and H is said a 1-strip if $|\mathcal{A}| = 1$, and a 2-strip if $|\mathcal{A}| = 2$.*

We often abuse notations and when we refer to a vertex of a strip (or a stable set etc.) we indeed consider a vertex (or a stable set etc.) of the graph in the strip.

Definition 3.2. *Let $\mathcal{G} = (G^1, \mathcal{A}^1), \ldots, (G^k, \mathcal{A}^k)$ be a family of k vertex disjoint strips, and let $\mathcal{P}$ be a partition of the multi-set of the cliques in $\mathcal{A}^1 \cup \ldots \cup \mathcal{A}^k$. The* composition *of the k strips w.r.t. $\mathcal{P}$ is the graph G that is obtained from the union of $G^1, \ldots, G^k$, by making adjacent vertices of $A \in \mathcal{A}^i$ and $B \in \mathcal{A}^j$ (i, j not necessarily different) if and only if A and B are in the same class of the partition $\mathcal{P}$. In this case we also say that $(\mathcal{G}, \mathcal{P})$, where $\mathcal{G} = \{(G^j, \mathcal{A}^j), j \in [k]\}$, defines a* strip decomposition *of G.*

We say that a graph G is *strip-composed* if G is a composition of some set of strips w.r.t. some partition $\mathcal{P}$. Each class of the partition of the extremities $\mathcal{P}$ defines a clique of the composed graph, and is called a *partition-clique*.

As an example we may think that every graph is strip-composed, with the simple strip decomposition where every strip is a vertex and eventually we take two times the same vertex in order to produce a 2-strip. It is easy to see that this construction is not feasible for every graph and in particular it is not feasible if the graph is not line.

Faenza, Oriolo and Stauffer in [17] have observed that the composition operation preserves some graph properties.

A strip $(G, \mathcal{A})$ is claw-free/quasi-line/line if the graph G_+ that is obtained from G as follows:

- if $H = (G, \mathcal{A})$ is a 2-strip, with $\mathcal{A} = \{A_1, A_2\}$, add two additional vertices a_1, a_2 such that $N(a_i) = A_i$, for $i = 1, 2$;

- if $H = (G, \mathcal{A})$ is a 1-strip, with $\mathcal{A} = \{A_1\}$, add one additional vertex a_1 such that $N(a_1) = A_1$,

is claw-free/quasi-line/line.

Lemma 3.3. *[17] The composition of claw-free/quasi-line/line strips is a claw-free/quasi-line/line graph.*

Suppose we are given a graph G and its strip decomposition $(\mathcal{G}, \mathcal{P})$. We would like to exploit this decomposition in order to solve some combinatorial optimization problems on G. One is clearly tempted to solve the problem on each strip, because it may be easier than solve the problem on the whole graph, and then try to combine the solutions obtained on each strip in order to obtain a solution for the problem on G. This idea is a little bit too naïve, but for the MWSS a slightly more complicated decomposition approach has shown to be successful.

3.2 The MWSS problem on strip-composed graphs

In this section we briefly describe one of the results proposed in [34], concerning the computation of the MWSS in a strip-composed graph G, given its strip decomposition $(\mathcal{G}, \mathcal{P})$. The main result of our interest in this chapter is the following (rephrased with our notation):

Theorem 3.4. *[34] Let $G(V, E)$ be the composition of strips $H_i = (G^i, \mathcal{A}^i)$ $i = 1 \ldots, k$ w.r.t. a partition $\mathcal{P}$ and let w be a non-negative weight function defined on the vertex set V. Suppose that for each $i = 1, \ldots, k$ we can compute a MWSS of H_i in time $O(p_i(|V|))$. Then the MWSS problem on G can be solved in time $O(\sum_{i=1}^{k} p_i(|V|) + match(|V|))$, where $match(n)$ is the time required to solve the matching problem on a graph with n vertices. If $p_i(|V|)$ is polynomial for each i, then the MWSS can be solved on G in polynomial time.*

We introduced at the beginning of the chapter that our aim is to prove a similar result for the MWCC on strip-composed perfect graphs, so before doing this we need to analyze the main steps that in [34] have brought to Theorem 3.4.

Every strip H_i is replaced with a suitable *gadget strip* T_i, that is a single vertex for each 1-strip and the complete graph with three vertices for each 2-strip (in this second case the extremities are two different edges of the triangle and not by chance both gagdets are line strips). Then they define a weight function on the vertices of those simpler strips; for every strip H_i with extremities A_1^i and A_2^i this function depends on the values $\alpha_w(G^i)$, $\alpha_w(G^i \setminus A_1^i)$, $\alpha_w(G^i \setminus A_2^i)$, $\alpha_w(G^i \setminus (A_1^i \cup A_2^i))$ and $\alpha_w(G^i \setminus (A_1^i \Delta A_2^i))$, thus if one can compute a MWSS of G^i in polynomial time one can compute the weight function of the simpler strips in polynomial time.

Together with the gadgets they define a suitable partition $\tilde{\mathcal{P}}$ of the extremities of the simpler strips. In this way they obtain a graph $\tilde{G}$ which is the strip composition of T_i $i = 1 \dots, k$ w.r.t. the partition $\tilde{\mathcal{P}}$, and this graph is line. Moreover from the construction of the gadget strips it is easy to translate a MWSS of $\tilde{G}$ into a MWSS of G.

Finally as $\tilde{G}$ is a line-graph they can find a MWSS of $\tilde{G}$ in the following way: they build the root graph of $\tilde{G}$ and they find a maximum weight matching in this graph. The solution of the matching problem on the root graph of $\tilde{G}$ basically gives the solution to the question that was under the naïve approach described before: how we suitably combine the MWSS of the strips in order to obtain a MWSS of G?

We underline that this result applies to all strip-composed graphs, while we aim at stating a similar result for the MWCC only for strip-composed *perfect* graphs. Moreover this result has been used as a building block for the first algorithm for the MWSS on claw-free graphs not based on augmenting paths techniques.

3.3 Sketch of the steps

In this section we want to give a big picture of the steps we need to obtain our algorithmic theorem for the MWCC in strip-composed perfect graphs. We give a short description of each step and we give a reminder to the appropriate sections for the details.

Suppose we are given a perfect graph G that is the composition of strips $H_1 = (G^1, \mathcal{A}^1), \dots, H_k = (G^k, \mathcal{A}^k)$ w.r.t. a partition $\mathcal{P}$, and a non-negative weight function w on $V(G)$. We will follow the approach outlined in the previous section for the MWSS; however, as we explain in the following, the task is now much more challenging.

We will compute a MWCC of G in three steps:

Step 1. We replace each strip H_i by a simple *gadget strip* $\tilde{H}_i = (\tilde{G}^i, \tilde{\mathcal{A}}^i)$ and compose the strips $\tilde{H}_i$ with respect to a suitable partition of the multiset $\bigcup_{i=1..k} \tilde{\mathcal{A}}^i$ as to obtain a graph $\tilde{G}$. However, we cannot use the gadgets strips defined in the previous section, as the graph $\tilde{G}$ might be imperfect: this will lead us to define four different new gadgets, with different parity properties, that are such that $\tilde{G}$ is odd hole free and line, thus perfect [43]. We also define a suitable weight function $\tilde{w}$ on the vertices of $\tilde{G}$, as well as new weight functions $w^1, \dots, w^k$ on the vertices of each strip. The details of this step are given in Section 3.5.

Step 2. Following [43], we may find a MWCC of $\tilde{G}$, w.r.t. the weight $\tilde{w}$, by running a primal-dual algorithm for the maximum weighted matching [19] on the root graph of $\tilde{G}$. The details of this Step are given in Section 3.6.

Step 3. We reconstruct a MWCC of G from a MWCC of $\tilde{G}$ and a MWCC of each of the strips H_i w.r.t. the weight function w^i. Again, this will be more involved than for the MWSS problem, because unfortunately there is not always a direct correspondence between cliques of $\tilde{G}$ and cliques of G. Moreover, for some 2-strips $H_i = (G^i, \mathcal{A}^i)$, besides a MWCC of the strip, we will also need also to compute a MWCC of some auxiliary graph associated to the strip: the graph $G^i_\bullet$ that is obtained from G^i by adding a new vertex x complete to both A^i_1 and A^i_2 and the graph $G^i_=$ that is the graph obtained from G^i by making A^i_1 complete to A^i_2. This is the more technical step, and all the details are given in Section 3.7.

3.4 Main result

Now we are ready to state the main result we want to prove in this chapter. Let $H_i = (G^i, \mathcal{A}^i)$ be a 2-strip, we indicate with $G^i_\bullet$ the graph obtained from G^i by adding a new vertex x complete to both A^i_1 and A^i_2 and with $G^i_=$ the graph obtained from G^i making A^i_1 complete to A^i_2. It will be clear shortly what we mean for even-short and odd-short strips; by now it is sufficient to know that they are subclasses of strips that can be easily recognized.

Theorem 3.5. *Let G be a perfect graph, composition of strips $H_i = (G^i, \mathcal{A}^i)$ $i = 1, \dots, k$ w.r.t. a partition $\mathcal{P}$. For each $i = 1, \dots, k$ let $O(p_i(|V(G^i)|))$ be the time required to compute:*

- *a* MWCC *of G^i and of $G^i_\bullet$, if H_i is an odd-short strip and $G^i_\bullet$ is an induced subgraph of G (thus perfect);*
- *a* MWCC *of G^i and of $G^i_=$, if $G^i_=$ is an induced subgraph of G (thus perfect), A^i_1 and in A^i_2 belong to the same class of $\mathcal{P}$, and there is an A_1–A_2 path of length two in the strip. In this case, when solving the* MWCC *on $G^i_=$, one can restrict to the case where the weight function w^i defined on $V(G^i_=)$ is such that $\alpha_{w^i}(G^i_=) = \alpha_{w^i}(G^i_= \setminus (A^i_1 \cup A^i_2))$;*
- *a* MWCC *of G^i else.*

Then the MWCC *problem on G can be solved in time $O(\sum_{i=1}^k p_i(|V(G^i)|) + match(|V(G)|))$, where $match(n)$ is the time required to solve the matching problem on a graph with n vertices. If $p_i(|V(G^i)|)$ is polynomial for each i, then the* MWCC *can be solved on G in polynomial time.*

In order to prove Theorem 3.5 we follow the steps outlined in Section 3.3.

In order to obtain the complexity bound required by Theorem 3.5 we need to compute in $O(p_i(|V|))$ the weight functions $\tilde{w}$ and w^i for $\tilde{G}$ and each strip H_i respectively, and we will see why this is possible; we will go in further details on this in Section 3.5. Moreover, each gadget is designed in order to produce a graph $\tilde{G}$ (the composition of $\tilde{H}_1, \dots, \tilde{H}_k$ with respect to $\tilde{\mathcal{P}}$), which is line and perfect. So, we also need to show that we can compute a MWCC of a graph which is line and perfect in time $O(match(|V|))$. In order to do this we again move to the root graph of $\tilde{G}$, and we will see in

Section 3.6, that solving the MWCC in $\tilde{G}$ is essentially equivalent to solving the dual of a maximum weight matching problem on the root graph of $\tilde{G}$.

Now we show that if every step of Section 3.3 is correct, the statement of our theorem is correct. Every gadget has at most three vertices, thus we can replace every strip in time $O(k) = O(|V(G)|)$. Moreover we have to compute at most four MWCC for each strip, thus this step takes $O(\sum_{i=1}^{k} p_i(|V|))$. Step 2 instead requires to compute the root graph of a line graphs with at most $3k$ vertices, and this can be done in time $O(k)$ (for a proof of this see [17]), and the dual of a maximum weight matching of H, therefore it can be done in time $O(match(|V|))$. Finally Step 3 requires to solve a MWCC for each strip (or eventually of some special graphs obtained from the strips), and it can be done again in time $O(\sum_{i=1}^{k} p_i(|V|))$. So we can conclude that if we can correctly perform all the four steps, we can compute a MWCC of G in time $O(\sum_{i=1}^{k} p_i(|V|) + match(|V|))$.

3.5 The gadgets

In this section we describe the gadgets we use for every strip. We also show that we can give an appropriate weight function to the vertices of those gadgets. We remark that we assume that we can find in time $O(p_i(|V|))$ a MWCC of the i-th strip, and thus we can compute the desired weight function of the i-th strip in time $O(p_i(|V|))$.

In this section we make a heavy use of duality between the MWCC and the MWSS. The fact that for every induced subgraph J of G, $\alpha_w(J) = \tau_w(J)$ is due to the perfection of G. We use this relation to easily prove the correctness of the weight function defined on the vertices of each gadget, but we underline that we never require the computation of any MWSS on the strips.

We denote the extremities of the strip H_i by $\mathcal{A}^i = \{A_1^i, A_2^i\}$ if H_i is a 2-strip and by $\mathcal{A}^i = \{A_1^i\}$ if H_i is a 1-strip. We now delve into three cases: (i) $H_i = (G^i, \mathcal{A}^i)$ is a 1-strip; (ii) $H_i = (G^i, \mathcal{A}^i)$ is a 2-strip with the extremities in the same class of the partition $\mathcal{P}$; (iii) $H_i = (G^i, \mathcal{A}^i)$ is a 2-strip with the extremities in different classes of the partition. However, we analyze the first two cases together.

For the first two cases the gadget will be a single vertex. In particular we define the trivial 1-strip $\tilde{H}_i^0 = (T_0^i, \tilde{\mathcal{A}}_0^i)$, where the graph T_0^i consists on a single vertex c^i, and $\tilde{\mathcal{A}}_0^i = \{\{c^i\}\}$.

(i) Let $\delta_1^i = \alpha_w(G^i \setminus A_1^i)$. We define $\tilde{w}(c^i) = \alpha_w(G^i) - \delta_1^i$.

(ii) Let $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$. We define $\tilde{w}(c^i) = \max\{\alpha_w(G^i \setminus A_1^i), \alpha_w(G^i \setminus A_2^i), \alpha_w(G^i \setminus (A_1^i \triangle A_2^i))\} - \delta_1^i$.

Finally, if we use $\tilde{H}_i^0$ instead of H_i in the composition, the new partition is $\mathcal{P}' := (\mathcal{P} \setminus \{P\}) \cup \{(P \setminus \mathcal{A}^i) \cup \tilde{\mathcal{A}}^i\}$, where $P \in \mathcal{P}$ was the set containing $\mathcal{A}^i$.

Next we show that replacing a 1-strip or a 2-strip with both extremities in the same class of $\mathcal{P}$ makes the value of the MWSS drop of a quantity equal to δ_1^i.

Lemma 3.6. *Let G be the composition of strips $H_1 = (G^1, \mathcal{A}^1), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$, and let $w : V(G) \to \mathbb{R}^+$ be a weight function. Suppose that H_1 is either a 1-strip or a 2-strip with the extremities in the same class of the partition $\mathcal{P}$. Let G' be the composition of strips $\tilde{H}_1^0 = (T_0^1, \tilde{\mathcal{A}}_0^1), H_2 = (G^2, \mathcal{A}^2), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ previously defined. Let $w' : V(G') \to \mathbb{R}^+$ be the weight function defined as $w'(v) = w(v)$ for $v \in \bigcup_{i=2..k} V(G^i)$, and $w'(c^1) = \tilde{w}(c^1)$. Then $\alpha_w(G) = \alpha_{w'}(G') + \delta_1^1$.*

Proof. First we prove $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$. Let $A = A_1$ if H_1 is a 1-strip and $A = A_1 \cup A_2$ if H_1 is a 2-strip with the extremities in the same class of the partition $\mathcal{P}$. Since A is a complete set in G, we can partition the stable sets S of G in the following way:

1) $S \cap A = \emptyset$;

2) $|S \cap A| = 1$.

In case 1), we have that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \delta_1^1$, where the last inequality follows from the fact that S misses

A. Therefore, $w(S \cap (G \setminus G^1)) \geq w(S) - \delta_1^1$. Moreover, $S \cap (G \setminus G^1)$ is a stable set of G' and $w'(S \cap (G \setminus G^1)) = w(S \cap (G \setminus G^1))$. It follows that $\alpha_{w'}(G') \geq w(S) - \delta_1^1$.

In case 2), we have that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1)$. If H_1 is a 1-strip, then $w(S \cap G^1) \leq \alpha_w(G^1)$. If H_1 is a 2-strip, then, as $|S \cap A| = 1$, we have that either $S \cap A \subseteq A_1^1 \cap A_2^1$, or $S \cap A \subseteq A_1^1 \setminus A_2^1$, or $S \cap A \subseteq A_2^1 \setminus A_1^1$. Then, either $S \cap G^1 \subseteq G^1 \setminus (A_1^1 \triangle A_2^1)$, or $S \cap G^1 \subseteq G^1 \setminus A_2^1$, or $S \cap G^1 \subseteq G^1 \setminus A_1^1$. So, $w(S \cap G^1) \leq \max\{\alpha_w(G^1 \setminus A_1^1), \alpha_w(G^1 \setminus A_2^1), \alpha_w(G^1 \setminus (A_1^1 \triangle A_2^1))\}$. In this case, $S \cap (G \setminus G^1) \cup \{c^1\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{c^1\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(c^1)$. Then we have that $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(c^1) = w(S) - w(S \cap G^1) + \tilde{w}(c^1) \geq w(S) - \delta_1^1$, where the last inequality holds by the previous case analysis.

Thus we have shown that for every stable set S of G, $\alpha_{w'}(G') \geq w(S) - \delta_1^1$. In particular, this must hold for a MWSS of G, so we obtain $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$.

Now we want to prove $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. We can partition the stable sets S' of G' in the following way:

1) $c^1 \notin S'$;

2) $c^1 \in S'$.

In case 1), let S^1 be a MWSS of $G^1 \setminus A$. Then, as S' misses c^1 and there are no edges between $G^1 \setminus A$ and $G \setminus G^1$, $S^1 \cup S'$ is a stable set of G. It follows that $\alpha_w(G) \geq w(S' \cup S^1) = w'(S') + \delta_1^1$.

In case 2), let S^1 be a stable set of G^1 having only one vertex in A of maximum weight. Now, $S' \setminus \{c^1\} \cup S^1$ is a stable set of G, so it holds $\alpha_w(G) \geq w(S' \setminus \{c^1\} \cup S^1) = w'(S') - \tilde{w}(c^1) + w(S^1)$. If H_1 is a 1-strip, then $w(S^1) = \alpha_w(G^1)$. If H_1 is a 2-strip, then $w(S^1) = \max\{\alpha_w(G^1 \setminus A_1^1), \alpha_w(G^1 \setminus A_2^1), \alpha_w(G^1 \setminus (A_1^1 \triangle A_2^1))\}$. In both cases, $w(S^1) - \tilde{w}(c^1) = \delta_1^1$, so $\alpha_w(G) \geq w'(S') + \delta_1^1$.

Thus we have shown that for every stable set S' of G', $\alpha_w(G) \geq w'(S') + \delta_1^1$. In particular, this must hold for a MWSS of G', so we obtain $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. $\square$

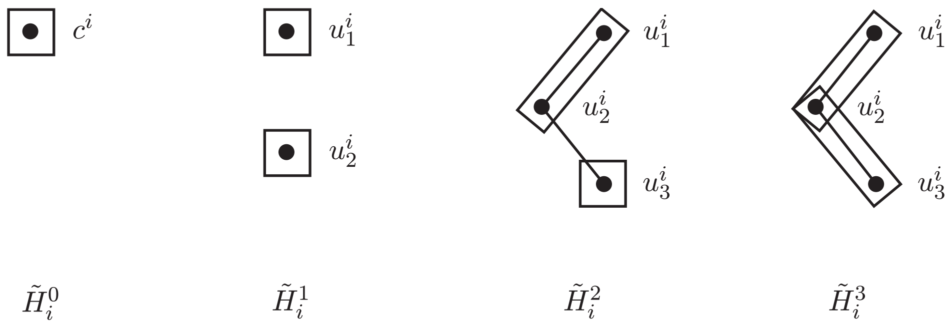


Figure 3.1: Trivial strips $\tilde{H}_i^0, \tilde{H}_i^1, \tilde{H}_i^2, \tilde{H}_i^3$, possibly associated with the strip H_i.

We will analyze now the case in which we have a 2-strip with the extremities in different classes of the partition. This case is more complicated because we need to take into consideration the parity of the strips. First we classify those 2-strips in even, odd, even-odd or non-connected.

Definition 3.7. *Let $U, W \subseteq V(G)$, we call a path $P = v_1, \ldots, v_k$ $(k \geq 1)$ a U–W path, if $v_1 \in U$, $v_k \in W$ and $v_i \notin U \cup W$ for $2 \leq i \leq k-1$.*

Definition 3.8. *We say that a 2-strip H_i is* non-connected *if there is no A_1^i–A_2^i path, and* connected *otherwise.*

Definition 3.9. *We say that a connected 2-strip H_i is* even *(resp.* odd*) if every A_1^i–A_2^i induced path has even (resp. odd) length. If a 2-strip has both even and odd length A_1^i–A_2^i induced paths, then we say that H_i is an* even-odd *strip.*

We call an odd or even-odd strip H_i *odd-short* if every odd A_1^i–A_2^i path has length one, and we call an even or even-odd strip H_i *even-short* if every even A_1^i–A_2^i path has length zero (i.e., it consists of a vertex in $A_1^i \cap A_2^i$).

In the following when speaking of an even (resp. odd) A_1^i–A_2^i path, we will always mean that the path is induced.

In order to preserve perfection of the original graph G, we have to use gadgets that have the same parity of the strips, otherwise we may introduce odd holes. So we have to introduce separately odd gadgets and even gadgets, plus some other convenient gadgets.

Let us consider a 2-strip $H_i = (G^i, \mathcal{A}^i)$ with the extremities in different classes of the partition $\mathcal{P}$. We want to introduce a gadget $\tilde{H}_i = (\tilde{G}^i, \tilde{\mathcal{A}}^i)$ and a new weight function $\tilde{w}$ on the vertices of $\tilde{G}^i$, in such a way that, when replacing H_i by $\tilde{H}_i$ in the strip composition for a suitable partition, the difference between the weights of the MWSS of the original graph and the MWSS of the new graph is δ_1^i, where $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$.

Definition 3.10. *Given a 2-strip $H_i = (G^i, \mathcal{A}^i)$ we define the associated trivial strip $\tilde{H}_i^1 = (T_1^i, \tilde{\mathcal{A}}_1^i)$ as follows: $V(T_1^i) = \{u_1^i, u_2^i\}$, $E(T_1^i) = \emptyset$, $\tilde{\mathcal{A}}_1^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i\}$, $\tilde{A}_2^i = \{u_2^i\}$. The new weight function $\tilde{w}$ gives the following weights to the vertices of T_1^i: $\tilde{w}(u_1^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, $\tilde{w}(u_2^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$.*

If we use $\tilde{H}_i^1$ instead of H_i in the composition, the new partition is $\mathcal{P}' := \mathcal{P} \setminus \{P_1, P_2\} \cup \{(P_1 \setminus \{A_1^i\}) \cup \{\{u_1^i\}\}, (P_2 \setminus \{A_2^i\}) \cup \{\{u_2^i\}\}\}$, where $P_1, P_2 \in \mathcal{P} : A_1^i \in P_1, A_2^i \in P_2$.

Definition 3.11. *Given a 2-strip $H_i = (G^i, \mathcal{A}^i)$ we define the associated trivial strip $\tilde{H}_i^2 = (T_2^i, \tilde{\mathcal{A}}_2^i)$ as the following graph: $V(T_2^i) = \{u_1^i, u_2^i, u_3^i\}$, $E(T_2^i) = \{u_1^i u_2^i, u_2^i u_3^i\}$, $\tilde{\mathcal{A}}_2^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i, u_2^i\}$, $\tilde{A}_2^i = \{u_3^i\}$. The new weight function $\tilde{w}$ gives the following weights to the vertices of T_2^i: $\tilde{w}(u_1^i) = \alpha_w(G^i) - \alpha_w(G^i \setminus A_1^i)$, $\tilde{w}(u_2^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, $\tilde{w}(u_3^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$.*

If we use $\tilde{H}_i^2$ instead of H_i in the composition, the new partition is $\mathcal{P}' := \mathcal{P} \setminus \{P_1, P_2\} \cup \{(P_1 \setminus \{A_1^i\}) \cup \{\{u_1^i, u_2^i\}\}, (P_2 \setminus \{A_2^i\}) \cup \{\{u_3^i\}\}\}$, where $P_1, P_2 \in \mathcal{P} : A_1^i \in P_1, A_2^i \in P_2$.

Definition 3.12. *Given a 2-strip $H_i = (G^i, \mathcal{A}^i)$ we define the associated trivial strip $\tilde{H}_i^3 = (T_3^i, \tilde{\mathcal{A}}_3^i)$ as the following graph: $V(T_3^i) = \{u_1^i, u_2^i, u_3^i\}$, $E(T_3^i) = \{u_1^i u_2^i, u_2^i u_3^i\}$, $\tilde{\mathcal{A}}_3^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i, u_2^i\}$, $\tilde{A}_2^i = \{u_2^i, u_3^i\}$. The new weight function $\tilde{w}$ gives the following weights to the vertices of T_3^i: $\tilde{w}(u_1^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, $\tilde{w}(u_2^i) = \alpha_w(G^i) - \delta_1^i$, $\tilde{w}(u_3^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$.*

If we use $\tilde{H}_i^3$ instead of H_i in the composition, the new partition is $\mathcal{P}' := \mathcal{P} \setminus \{P_1, P_2\} \cup \{(P_1 \setminus \{A_1^i\}) \cup \{\{u_1^i, u_2^i\}\}, (P_2 \setminus \{A_2^i\}) \cup \{\{u_2^i, u_3^i\}\}\}$, where $P_1, P_2 \in \mathcal{P} : A_1^i \in P_1, A_2^i \in P_2$.

In next lemmas we show that the given weights are correct, in the sense that every time we replace a 2-strip with the corresponding gadget the value of the MWSS drops by δ_1^i. These lemmas have a different proof depending on the fact that the relation $\alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) \gtreqless \alpha_w(G^1) + \delta_1^1$, is satified with $=$, $>$ or $<$. We will se later on that the satisfaction of this relation is strictly related to the parity of the strips.

Lemma 3.13. *Let G be the composition of strips $H_1 = (G^1, \mathcal{A}^1), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$, and let $w : V(G) \to \mathbb{R}^+$ be a weight function. Suppose that H_1 is a 2-strip with the extremities in different classes of the partition $\mathcal{P}$. Let G' be the composition of strips $\tilde{H}_1^1 = (T_1^1, \tilde{\mathcal{A}}_1^1), H_2 = (G^2, \mathcal{A}^2), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ previously defined. Let $w' : V(G') \to \mathbb{R}^+$ be the weight function defined as $w'(v) = w(v)$ for $v \in \bigcup_{i=2..k} V(G^i)$, and $w'(v) = \tilde{w}(v)$ for $v \in V(T_1^1)$. If $\alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) = \alpha_w(G^1) + \delta_1^1$, then $\alpha_w(G) = \alpha_{w'}(G') + \delta_1^1$.*

Proof. First we prove $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$. We can partition the stable sets S of G in the following way:

1) $S \cap (A_1^1 \cup A_2^1) = \emptyset$;

2) $|S \cap A_1^1| + |S \cap A_2^1| = 1$;

3) $|S \cap A_1^1| + |S \cap A_2^1| = 2$.

In case 1), we have that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \delta_1^1$, where the last inequality follows from the fact that S misses both A_1^1 and A_2^1. Therefore, $w(S \cap (G \setminus G^1)) \geq w(S) - \delta_1^1$. Moreover, $S \cap (G \setminus G^1)$ is a stable set of G' and $w'(S \cap (G \setminus G^1)) = w(S \cap (G \setminus G^1))$. It follows that $\alpha_{w'}(G') \geq w(S) - \delta_1^1$.

Now we analyze case 2), in particular we suppose that $|S \cap A_1^1| = 1$ and $|S \cap A_2^1| = 0$; it follows that $S \cap A_2^1 = \emptyset$. Then $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \alpha_w(G^1 \setminus A_2^1)$, where again the last inequality follows from the fact that S misses A_2^1. Now we observe that $S \cap (G \setminus G^1) \cup \{u_1^1\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{u_1^1\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1)$ so we have that $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1) = w(S \cap (G \setminus G^1)) + \alpha_w(G^1 \setminus$

$A_2^1) - \delta_1^1 \geq w(S) - \delta_1^1$. The case in which $|S \cap A_2^1| = 1$ and $|S \cap A_1^1| = 0$ is analogous.

In case 3), we have that both $|S \cap A_1^1| = 1$ and $|S \cap A_2^1| = 1$, that is if $A_1^1 \cap A_2^1 = \emptyset$, S takes a vertex in A_1^1 and a vertex in A_2^1, else S may take a single vertex in $A_1^1 \cap A_2^1$ or a vertex in $A_1^1 \setminus A_2^1$ and a vertex in $A_2^1 \setminus A_1^1$. It follows that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \alpha_w(G^1)$. Moreover, $S \cap (G \setminus G^1) \cup \{u_1^1, u_2^1\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{u_1^1, u_2^1\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1) + \tilde{w}(u_2^1)$. Then we have that $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1) + \tilde{w}(u_2^1) = w(S \cap (G \setminus G^1)) + \alpha_w(G^1 \setminus A_2^1) - \delta_1^1 + \alpha_w(G^1 \setminus A_1^1) - \delta_1^1 = w(S \cap (G \setminus G^1)) + \alpha_w(G^1) - \delta_1^1 \geq w(S) - \delta_1^1$, where the last equality holds by hypothesis.

Thus we have shown that for every stable set S of G, $\alpha_{w'}(G') \geq w(S) - \delta_1^1$. In particular, this must hold for a MWSS of G, so we obtain $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$.

Now we want to prove $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. We can partition the stable sets S' of G' in the following way:

1) $S' \cap \{u_1^1, u_2^1\} = \emptyset$;

2) $|S' \cap \{u_1^1, u_2^1\}| = 1$;

3) $|S' \cap \{u_1^1, u_2^1\}| = 2$.

In case 1), let S^1 be a MWSS of $G^1 \setminus (A_1^1 \cup A_2^1)$. Then, as S' misses both u_1^1 and u_2^1 and there are no edges between $G^1 \setminus (A_1^1 \cup A_2^1)$ and $G \setminus G^1$, $S^1 \cup S'$ is a stable set of G. It follows that $\alpha_w(G) \geq w(S' \cup S^1) = w'(S') + \delta_1^1$.

In case 2), we suppose that $u_1^1 \in S'$ and let S^1 be a MWSS of $G^1 \setminus A_2^1$. Then, as S' misses u_2^1, $S' \setminus \{u_1^1\} \cup S^1$ is a stable set of G. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_1^1\} \cup S^1) = w'(S') - \tilde{w}(u_1^1) + \alpha_w(G^1 \setminus A_2^1) = w'(S') - \alpha_w(G^1 \setminus A_2^1) + \delta_1^1 + \alpha_w(G^1 \setminus A_2^1) = w'(S') + \delta_1^1$. The case in which $u_2^1 \in S'$ is analogous.

In case 3), let S^1 be a MWSS of G^1. Now, $S' \setminus \{u_1^1, u_2^1\} \cup S^1$ is a stable set of G, so it holds $\alpha_w(G) \geq w(S' \setminus \{u_1^1, u_2^1\} \cup S^1) = w'(S') - \tilde{w}(u_1^1) - \tilde{w}(u_2^i) + \alpha_w(G^1) = w'(S') - \alpha_w(G^1 \setminus A_1^1) - \alpha_w(G^1 \setminus A_2^1) + 2\delta_1^1 + \alpha_w(G^1) = w'(S') + \delta_1^1$, where the last equality holds by hypothesis.

Thus we have shown that for every stable set S' of G', $\alpha_w(G) \geq w'(S') + \delta^1_1$. In particular, this must hold for a MWSS of G', so we obtain $\alpha_w(G) \geq \alpha_{w'}(G') + \delta^1_1$. $\square$

Lemma 3.14. *Let G be the composition of strips $H_1 = (G^1, \mathcal{A}^1), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$, and let $w : V(G) \to \mathbb{R}^+$ be a weight function. Suppose that H_1 is a 2-strip with the extremities in different classes of the partition $\mathcal{P}$. Let G' be the composition of strips $\tilde{H}^2_1 = (T^1_2, \tilde{\mathcal{A}}^1_2), H_2 = (G^2, \mathcal{A}^2), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ previously defined. Let $w' : V(G') \to \mathbb{R}^+$ be the weight function defined as $w'(v) = w(v)$ for $v \in \bigcup_{i=2..k} V(G^i)$, and $w'(v) = \tilde{w}(v)$ for $v \in V(T^1_2)$. If $\alpha_w(G^1 \setminus A^1_1) + \alpha_w(G^1 \setminus A^1_2) \geq \alpha_w(G^1) + \delta^1_1$, then $\alpha_w(G) = \alpha_{w'}(G') + \delta^1_1$.*

Proof. First we prove $\alpha_w(G) \leq \alpha_{w'}(G') + \delta^1_1$. We can partition the stable sets S of G in the following way:

1) $S \cap (A^1_1 \cup A^1_2) = \emptyset$;

2) $|S \cap A^1_1| + |S \cap A^1_2| = 1$;

3) $|S \cap A^1_1| + |S \cap A^1_2| = 2$.

In case 1), we have that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \delta^1_1$, where the last inequality follows from the fact that S misses both A^1_1 and A^1_2. Therefore, $w(S \cap (G \setminus G^1)) \geq w(S) - \delta^1_1$. Moreover, $S \cap (G \setminus G^1)$ is a stable set of G' and $w'(S \cap (G \setminus G^1)) = w(S \cap (G \setminus G^1))$. It follows that $\alpha_{w'}(G') \geq w(S) - \delta^1_1$.

Now we analyze case 2). First, suppose that $|S \cap A^1_1| = 1$ and $|S \cap A^1_2| = 0$ (or in other words S misses A^1_2). Then $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \alpha_w(G^1 \setminus A^1_2)$, where again the last inequality follows from the fact that S misses A^1_2. Now observe that $S \cap (G \setminus G^1) \cup \{u^1_2\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{u^1_2\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(u^1_2)$. Then we have that $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u^1_2) \geq w(S) - \alpha_w(G^1 \setminus A^1_2) + \alpha_w(G^1 \setminus A^1_2) - \delta^1_1 = w(S) - \delta^1_1$. Now suppose that $|S \cap A^1_2| = 1$ and $|S \cap A^1_1| = 0$ (or in other words S misses A^1_1). We obtain $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1) + \alpha_w(G^1 \setminus A^1_1)$. In this case, $S \cap (G \setminus G^1) \cup \{u^1_3\}$ is a stable set of

G' and this gives rise to the inequality $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u_3^1) \geq w(S) - \alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_1^1) - \delta_1^1 = w(S) - \delta_1^1$.

In case 3), we have that both $|S \cap A_1^1| = 1$ and $|S \cap A_2^1| = 1$, that is if $A_1^1 \cap A_2^1 = \emptyset$, S takes a vertex in A_1^1 and a vertex in A_2^1, else S may take a single vertex in $A_1^1 \cap A_2^1$ or a vertex in $A_1^1 \setminus A_2^1$ and a vertex in $A_2^1 \setminus A_1^1$. It follows that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \alpha_w(G^1)$. Moreover, $S \cap (G \setminus G^1) \cup \{u_1^1, u_3^1\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{u_1^1, u_3^1\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1) + \tilde{w}(u_3^1)$. Then we have that $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1) + \tilde{w}(u_3^1) \geq w(S) - \alpha_w(G^1) + \alpha_w(G^1) - \delta_1^1 = w(S) - \delta_1^1$.

Thus we have shown that for every stable set S of G, $\alpha_{w'}(G') \geq w(S) - \delta_1^1$. In particular, this must hold for a MWSS of G, so we obtain $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$.

Now we want to prove $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. We can partition the stable sets S' of G' in the following way:

1) $S' \cap \{u_1^1, u_2^1, u_3^1\} = \emptyset$;

2) $|S' \cap \{u_1^1, u_2^1, u_3^1\}| = 1$;

3) $|S' \cap \{u_1^1, u_2^1, u_3^1\}| = 2$.

In case 1), let S^1 be a MWSS of $G^1 \setminus (A_1^1 \cup A_2^1)$. Then, as S' misses u_1^1, u_2^1 and u_3^1, and there are no edges between $G^1 \setminus (A_1^1 \cup A_2^1)$ and $G \setminus G^1$, $S^1 \cup S'$ is a stable set of G. It follows that $\alpha_w(G) \geq w(S' \cup S^1) = w'(S') + \delta_1^1$.

In case 2), first suppose that $u_1^1 \in S'$ and let S^1 be a MWSS of $G^1 \setminus A_2^1$. Then, as S' misses u_2^1 and u_3^1, $S' \setminus \{u_1^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_1^1\} \cup S^1) = w'(S') - \tilde{w}(u_1^1) + \alpha_w(G^1 \setminus A_2^1)$. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_1^1\} \cup S^1) = w'(S') - \alpha_w(G^1) + \alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) \geq w'(S') + \delta_1^1$, where the last inequality holds by hypothesis. Now suppose that $u_2^1 \in S'$ and let S^1 be a MWSS of $G^1 \setminus A_2^1$. Then, as S' misses u_1^1 and u_3^1, $S' \setminus \{u_2^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_2^1\} \cup S^1) = w'(S') - \tilde{w}(u_2^1) + \alpha_w(G^1 \setminus A_2^1)$. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_2^1\} \cup S^1) = w'(S') - \alpha_w(G^1 \setminus A_2^1) + \delta_1^1 + \alpha_w(G^1 \setminus A_2^1) = w'(S') + \delta_1^1$. Finally, suppose that $u_3^1 \in S'$ and let S^1 be a MWSS of $G^1 \setminus A_1^1$. Then, as S' misses u_1^1 and u_2^1, $S' \setminus \{u_3^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_3^1\} \cup S^1) = w'(S') - \tilde{w}(u_3^1) + \alpha_w(G^1 \setminus A_1^1)$. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_3^1\} \cup S^1) = w'(S') - \alpha_w(G^1 \setminus A_1^1) + \delta_1^1 + \alpha_w(G^1 \setminus A_1^1) = w'(S') + \delta_1^1$.

In case 3), from the structure of T_2, we have that $\{u_1^1, u_3^1\} \subseteq S'$. Let S^1 be a MWSS of G^1. Now, $S' \setminus \{u_1^1, u_3^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_1^1, u_3^1\} \cup S^1) = w'(S') - \tilde{w}(u_1^1) - \tilde{w}(u_3^1) + \alpha_w(G^1)$. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_1^1, u_2^1\} \cup S^1) = w'(S') - \alpha_w(G^1) + \delta_1^1 + \alpha_w(G^1) = w'(S') + \delta_1^1$.

Thus we have shown that for every stable set S' of G', $\alpha_w(G) \geq w'(S') + \delta_1^1$. In particular, this must hold for a MWSS of G', so we obtain $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. □

Lemma 3.15. *Let G be the composition of strips $H_1 = (G^1, \mathcal{A}^1), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$, and let $w : V(G) \to \mathbb{R}^+$ be a weight function. Suppose that H_1 is a 2-strip with the extremities in different classes of the partition $\mathcal{P}$. Let G' be the composition of strips $\tilde{H}_1^3 = (T_3^1, \tilde{\mathcal{A}}_3^1), H_2 = (G^2, \mathcal{A}^2), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ previously defined. Let $w' : V(G') \to \mathbb{R}^+$ be the weight function defined as $w'(v) = w(v)$ for $v \in \bigcup_{i=2..k} V(G^i)$, and $w'(v) = \tilde{w}(v)$ for $v \in V(T_3^1)$. If $\alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) \leq \alpha_w(G^1) + \delta_1^1$, then $\alpha_w(G) = \alpha_{w'}(G') + \delta_1^1$.*

Proof. First we prove $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$. We can partition the stable sets S of G in the following way:

1) $S \cap (A_1^1 \cup A_2^1) = \emptyset$;

2) $|S \cap A_1^1| + |S \cap A_2^1| = 1$;

3) $|S \cap A_1^1| + |S \cap A_2^1| = 2$.

In case 1), we have that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \delta_1^1$, where the last inequality follows from the fact that S misses both A_1^1 and A_2^1. Therefore, $w(S \cap (G \setminus G^1)) \geq w(S) - \delta_1^1$. Moreover, $S \cap (G \setminus G^1)$ is a stable set of G' and $w'(S \cap (G \setminus G^1)) = w(S \cap (G \setminus G^1))$. It follows that $\alpha_{w'}(G') \geq w(S) - \delta_1^1$.

Now we analyze case 2), in particular we suppose that $|S \cap A_1^1| = 1$ and $|S \cap A_2^1| = 0$ (that is S misses A_2^1). Then $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \alpha_w(G^1 \setminus A_2^1)$, where again the last inequality follows from the fact that S misses A_2^1. Now we observe that $S \cap (G \setminus G^1) \cup \{u_1^1\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{u_1^1\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1)$. We have that

$\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u_1^1) \geq w(S) - \alpha_w(G^1 \setminus A_2^1) + \alpha_w(G^1 \setminus A_2^1) - \delta_1^1 = w(S) - \delta_1^1$. The case where $|S \cap A_2^1| = 1$ and $|S \cap A_1^1| = 0$ goes along the same lines.

In case 3), we have that both $|S \cap A_1^1| = 1$ and $|S \cap A_2^1| = 1$, that is if $A_1^1 \cap A_2^1 = \emptyset$, S takes a vertex in A_1^1 and a vertex in A_2^1, else S may take a single vertex in $A_1^1 \cap A_2^1$ or a vertex in $A_1^1 \setminus A_2^1$ and a vertex in $A_2^1 \setminus A_1^1$. It follows that $w(S) = w(S \cap (G \setminus G^1)) + w(S \cap G^1) \leq w(S \cap (G \setminus G^1)) + \alpha_w(G^1)$. Moreover, $S \cap (G \setminus G^1) \cup \{u_2^1\}$ is a stable set of G', and $w'(S \cap (G \setminus G^1) \cup \{u_2^1\}) = w(S \cap (G \setminus G^1)) + \tilde{w}(u_2^1)$. Then we have that $\alpha_{w'}(G') \geq w(S \cap (G \setminus G^1)) + \tilde{w}(u_2^1) \geq w(S) - \alpha_w(G^1) + \alpha_w(G^1) - \delta_1^1 = w(S) - \delta_1^1$.

Thus we have shown that for every stable set S of G, $\alpha_{w'}(G') \geq w(S) - \delta_1^1$. In particular, this must hold for a MWSS of G, so we obtain $\alpha_w(G) \leq \alpha_{w'}(G') + \delta_1^1$.

Now we want to prove $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. We can partition the stable sets S' of G' in the following way:

1) $S' \cap \{u_1^1, u_2^1, u_3^1\} = \emptyset$;

2) $|S' \cap \{u_1^1, u_2^1, u_3^1\}| = 1$;

3) $|S' \cap \{u_1^1, u_2^1, u_3^1\}| = 2$.

In case 1), let S^1 be a MWSS of $G^1 \setminus (A_1^1 \cup A_2^1)$. Then, as S' misses u_1^1, u_2^1 and u_3^1, and there are no edges between $G^1 \setminus (A_1^1 \cup A_2^1)$ and $G \setminus G^1$, $S^1 \cup S'$ is a stable set of G. It follows that $\alpha_w(G) \geq w(S' \cup S^1) = w'(S') + \delta_1^1$.

In case 2), first suppose that $u_1^1 \in S'$ and let S^1 be a MWSS of $G^1 \setminus A_2^1$. Then, as S' misses u_2^1 and u_3^1, $S' \setminus \{u_1^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_1^1\} \cup S^1) = w'(S') - \tilde{w}(u_1^1) + \alpha_w(G^1 \setminus A_2^1)$. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_1^1\} \cup S^1) = w'(S') - \alpha_w(G^1 \setminus A_2^1) + \delta_1^1 + \alpha_w(G^1 \setminus A_2^1) = w'(S') + \delta_1^1$. The case where $u_3^1 \in S'$ goes along the same lines. Finally, let us suppose that $u_2^1 \in S'$ and let S^1 be a MWSS of G^1. Then $S' \setminus \{u_2^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_2^1\} \cup S^1) = w'(S') - \tilde{w}(u_2^1) + \alpha_w(G^1)$. It follows that $\alpha_w(G) \geq w(S' \setminus \{u_2^1\} \cup S^1) = w'(S') - \alpha_w(G^1) + \delta_1^1 + \alpha_w(G^1) = w'(S') + \delta_1^1$.

In case 3), from the structure of T_2, we have that $\{u_1^1, u_3^1\} \subseteq S'$. Let S^1 be a MWSS of G^1. Now, $S' \setminus \{u_1^1, u_3^1\} \cup S^1$ is a stable set of G, and $w(S' \setminus \{u_1^1, u_3^1\} \cup S^1) = w'(S') - \tilde{w}(u_1^1) - \tilde{w}(u_3^1) + \alpha_w(G^1)$. It follows that

$\alpha_w(G) \geq w(S' \setminus \{u_1^1, u_2^1\} \cup S^1) = w'(S') - \alpha_w(G^1 \setminus A_2^1) + 2\delta_1^1 - \alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1) \geq w'(S') + \delta_1^1$, where the last inequality holds by hypothesis.

Thus we have shown that for every stable set S' of G', $\alpha_w(G) \geq w'(S') + \delta_1^1$. In particular, this must hold for a MWSS of G', so we obtain $\alpha_w(G) \geq \alpha_{w'}(G') + \delta_1^1$. □

Here we show how the condition on the weights of the gadgets is related to the parity of the strips.

Lemma 3.16. *If $H_1 = (G^1, \mathcal{A}^1)$ is a non-connected strip then $\alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) = \alpha_w(G^1) + \delta_1^1$.*

Proof. Let G_1^1 be the connected component of G^1 that contains A_1^1, and let $G_2^1 = G^1 \setminus V(G_1^1)$. Since G^1 is a non-connected strip, then A_2^1 is contained in G_2^1. The following equalities are straightforward, and imply the lemma.

$$\begin{aligned}
\alpha_w(G^1) &= \alpha_w(G_1^1) + \alpha_w(G_2^1) \\
\alpha_w(G^1 \setminus A_2^1) &= \alpha_w(G_1^1) + \alpha_w(G_2^1 \setminus A_2^1) \\
\alpha_w(G^1 \setminus A_1^1) &= \alpha_w(G_1^1 \setminus A_1^1) + \alpha_w(G_2^1) \\
\delta_1^1 &= \alpha_w(G_1^1 \setminus A_1^1) + \alpha_w(G_2^1 \setminus A_2^1)
\end{aligned}$$

□

Lemma 3.17. *If $H_1 = (G^1, \mathcal{A}^1)$ is an odd strip and G^1 is perfect then $\alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) \geq \alpha_w(G^1) + \delta_1^1$.*

Proof. Let G^* be the graph obtained in the following way: add to G^1 a vertex v_1 complete to A_1^1, a vertex v_2 complete to A_2^1, with v_1 adjacent to v_2. As H_1 is odd, $A_1^1 \cap A_2^1 = \emptyset$. Besides, as G^1 is perfect, G^* is perfect. We want to extend w to v_1 and v_2, so we let $a = w(v_1)$ and $b = w(v_2)$. We choose $a, b \geq 0$ and such that $a + \alpha_w(G^1 \setminus A_1^1) = b + \alpha_w(G^1 \setminus A_2^1) > \alpha_w(G^1)$: that is always possible. A stable set in G^* can either take v_1 and then no vertex of A_1^1, or it can take v_2 and then no vertex of A_2^1 or it can miss both

v_1 and v_2. Then for our choice of the weights of v_1 and v_2 we have that $\alpha_w(G^*) = a + \alpha_w(G^1 \setminus A_1^1) = b + \alpha_w(G^1 \setminus A_2^1)$.

Let y be a MWCC for (G^*, w) and let us denote with $\tau_w(G^*)$ its value; we call h_1 the value given by y to the clique $\{v_1\} \cup A_1^1$, that is $h_1 = y_{\{v_1\} \cup A_1^1}$, we call h_3 the value given by y to the clique $\{v_2\} \cup A_2^1$, that is $h_3 = y_{\{v_2\} \cup A_2^1}$, and we call h_2 the value given by y to the clique $\{v_1, v_2\}$, that is $h_2 = y_{\{v_1, v_2\}}$. Now we define a new weight function $\tilde{w}$ on $V(G^1)$: $\tilde{w}(v) = w(v)$ $\forall v \in V(G^1) \setminus (A_1^1 \cup A_2^1)$, $\tilde{w}(v) = w(v) - h_1$ $\forall v \in A_1^1$, $\tilde{w}(v) = w(v) - h_3$ $\forall v \in A_2^1$. Let us denote with $\tau_{\tilde{w}}(G^1)$ the value of a MWCC of $(G^1, \tilde{w})$, then $\tau_w(G^*) = h_1 + h_2 + h_3 + \tau_{\tilde{w}}(G^1)$ by the optimality of y for G^* and the definition of h_1, h_2, h_3.

As G^* is perfect we know that the maximum weight stable set problem and the minimum weight clique cover on G^* are dual problems and so every vertex v belonging to a MWSS of G^* is covered exactly by a MWCC of G^*, that is $\sum_{C \in \mathcal{K}(G^*): v \in C} y_C = w(v)$, where $\mathcal{K}(G^*)$ is the set of maximal cliques of G^*. In particular, for our choice of a and b, both v_1 and v_2 belong to MWSS of G^*, so we have that $h_1 + h_2 = a$ and $h_2 + h_3 = b$. Moreover, again by duality, $\alpha_w(G^*) = \tau_w(G^*)$, and we obtain $h_1 + h_2 + h_3 + \tau_{\tilde{w}}(G^1) = a + \alpha_w(G^1 \setminus A_1^1)$, that is $h_3 + \tau_{\tilde{w}}(G^1) = \alpha_w(G^1 \setminus A_1^1)$, and $h_1 + h_2 + h_3 + \tau_{\tilde{w}}(G^1) = b + \alpha_w(G^1 \setminus A_2^1)$, that is $h_1 + \tau_{\tilde{w}}(G^1) = \alpha_w(G^1 \setminus A_2^1)$. But again by duality and by the perfection of G^1 we can rewrite those two equations as (i) $h_3 + \alpha_{\tilde{w}}(G^1) = \alpha_w(G^1 \setminus A_1^1)$ and (ii) $h_1 + \alpha_{\tilde{w}}(G^1) = \alpha_w(G^1 \setminus A_2^1)$.

As A_1^1 and A_2^1 are cliques we can easily deduce the inequality $\alpha_w(G^1) \leq \alpha_{\tilde{w}}(G^1) + h_1 + h_3$ and by definition of the weight function $\tilde{w}$ it follows that $\delta_1^1 \leq \alpha_{\tilde{w}}(G^1)$; summing up these inequalities we obtain $\alpha_w(G^1) + \delta_1^1 \leq 2\alpha_{\tilde{w}}(G^1) + h_1 + h_3$, then using (i) and (ii) $\alpha_w(G^1) + \delta_1^1 \leq \alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1)$. □

Lemma 3.18. *If $H_1 = (G^1, \mathcal{A}^1)$ is an even strip and G^1 is perfect then $\alpha_w(G^1 \setminus A_1^1) + \alpha_w(G^1 \setminus A_2^1) \leq \alpha_w(G^1) + \delta_1^1$.*

Proof. We build the following auxiliary strip $H^* = (G^*, \mathcal{A}^*)$: we add a vertex v complete to A_1^1 and $\mathcal{A}^* = \{\{v\}, A_2^1\}$. We observe that by construction and the hypothesis on H_1, H^* is an odd strip and G^* is perfect. We extend the weight function of G^1 to v, putting $w(v) = a$, where $a > \alpha_w(G^1)$. From the choice of a we have the following equalities:

$$\begin{array}{rcl} \alpha_w(G^* \setminus \{v\}) & = & \alpha_w(G^1) \\ \alpha_w(G^* \setminus (\{v\} \cup A_2^1)) & = & \alpha_w(G^1 \setminus A_2^1) \\ \alpha_w(G^*) \; = \; \max\{a + \alpha_w(G^1 \setminus A_1^1), \alpha_w(G^1)\} & = & a + \alpha_w(G^1 \setminus A_1^1) \\ \alpha_w(G^* \setminus A_2^1) \; = \; \max\{a + \delta_1^1, \alpha_w(G^1 \setminus A_2^1)\} & = & a + \delta_1^1 \end{array}$$

By Lemma 3.17 the following inequality holds $\alpha_w(G^* \setminus \{v\}) + \alpha_w(G^* \setminus A_2^1) \geq \alpha_w(G^* \setminus (\{v\} \cup A_2^1)) + \alpha_w(G^*)$, that is $\alpha_w(G^1) + a + \delta_1^1 \geq \alpha_w(G^1 \setminus A_2^1) + a + \alpha_w(G^1 \setminus A_1^1)$ and therefore $\alpha_w(G^1) + \delta_1^1 \geq \alpha_w(G^1 \setminus A_2^1) + \alpha_w(G^1 \setminus A_1^1)$. □

As we have described more than one gadget we should give a method to choose one gadget for every 2-strip H_i. In particular, if we can calculate the values of the crucial clique covers ($\tau_w(G^i)$, $\tau_w(G^i \setminus A_2^i)$, $\tau_w(G^i \setminus A_1^i)$ and δ_1^i) for each strip we can determine whether one of these three relations holds

1. $\tau_w(G^i) + \tau_1^i > \tau_w(G^i \setminus A_2^i) + \tau_w(G^i \setminus A_1^i)$
2. $\tau_w(G^i) + \delta_1^i < \tau_w(G^i \setminus A_2^i) + \tau_w(G^i \setminus A_1^i)$
3. $\tau_w(G^i) + \delta_1^i = \tau_w(G^i \setminus A_2^i) + \tau_w(G^i \setminus A_1^i)$

If 1) holds we know that the strip is either even or even-odd and we can use $\tilde{H}_i^3$ as a suitable gadget. If 2) holds we know that the strip is either odd or even-odd and we can use $\tilde{H}_i^2$ as a suitable gadget. If 3) holds we can simply use $\tilde{H}_i^1$ as a suitable gadget. In this way we obtain an odd hole free graph, which is line and so perfect (odd antiholes with length greater than 7 are not line).

Remark 1. *If G composition of strips $H_1 = (G^1, \mathcal{A}^1), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$ is odd hole free then G' composition of $\tilde{H}_1^0 = (T_0^1, \tilde{\mathcal{A}}_0^1), H_2 = (G^2, \mathcal{A}^2), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ is odd hole free. If G composition of strips $H_1 = (G^1, \mathcal{A}^1), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$ is odd hole free then G' composition of $\tilde{H}_1^1 = (T_1^1, \tilde{\mathcal{A}}_1^1), H_2 = (G^2, \mathcal{A}^2), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ is odd hole free. If G composition of strips $H_1 = (G^1, \mathcal{A}^1), \dots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$ is odd hole free and H_1 is either an* *odd strip or an even-odd strip then G' composition of $\tilde{H}_1^2 = (T_2^1, \tilde{\mathcal{A}}_2^1), H_2 = (G^2, \mathcal{A}^2), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ is odd hole free. If G composition of strips $H_1 = (G^1, \mathcal{A}^1), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to a partition $\mathcal{P}$ is odd hole free and H_1 is either an even strip or an even-odd strip then G' composition of $\tilde{H}_1^3 = (T_3^1, \tilde{\mathcal{A}}_3^1), H_2 = (G^2, \mathcal{A}^2), \ldots, H_k = (G^k, \mathcal{A}^k)$ with respect to the partition $\mathcal{P}'$ is odd hole free.*

Remark 2. *Strips $\tilde{H}_i^0, \tilde{H}_i^1, \tilde{H}_i^2, \tilde{H}_i^3$ are line strips. So, if we iteratively make the substitution of the strip H_i by the corresponding gadget $\tilde{H}_i^j$, from Lemma 3.3 the graph we obtain is a line graph.*

Remark 3. *Line graphs are $\overline{C_k}$-free for $k \geq 7$.*

This last remark follows from the characterization of line graphs in terms of forbidden induced subgraphs given by Beineke [2]. From these two remarks it follows that if we iteratively substitute on G every strip with the corresponding gadget with respect to the validity of 1, 2 or 3, we obtain a graph $\tilde{G}$ that is odd-hole free and a line graph, so it is perfect.

As a corollary of Lemmas 3.6, 3.13, 3.14 and 3.15, it follows that $\alpha_w(G) = \alpha_w(\tilde{G}) + \sum_{i=1}^{k} \delta_1^i$. Since both graphs are perfect, by duality the same relation holds for the values of the MWCC of the two graphs.

3.6 Weighted clique cover of line and perfect graphs

Once we have replaced every strip with the suitable gadget we end up with a graph $\tilde{G} = L(H)$ which is a line graph and it is perfect. We are then left with the problem of solving the MWCC on $L(H)$. In particular we can observe that maximal cliques in $L(H)$ correspond to multistars (i.e all the edges incident to a vertex v) and multitriangles (i.e. a complete graph on three vertices with eventually multiple edges) in the root graph H, so we can try to solve a weighted stars and triangles edge cover in H instead that a MWCC in $L(H)$.

We observe that for our purpose we may assume that H is simple; in fact if we have a set of parallel edges we delete all but one edge and we give to this edge the maximum weight among the edges in the set, instead if we have a loop we know that it corresponds to a simplicial vertex in $L(H)$, so we can cover it at the price of its weight with a suitable clique in $L(H)$. Consequently the LP program for the weighted stars and triangles cover in H is the following

$$\min \sum_{v \in V} y_v + \sum_{t \in T} \pi_t$$

$$y_u + y_v + \sum_{t \in T : uv \in t} \pi_t \geq w_{uv} \qquad \forall uv \in E(H) : u \neq v$$

$$y_v \geq 0 \qquad \forall v \in V(H)$$

$$\pi_t \geq 0 \qquad \forall t \in T$$

where T is the set of all the induced triangles of H. This linear program looks like the odd set cover problem, i.e. the dual of the maximum weight matching in H, except for the fact that we are not considering all the odd sets (or similarly all the 2-connected hypomatchable subgraphs) but just the triangles.

We can use the Edmonds' primal dual algorithm for the maximum weight matching to obtain a solution to this problem. We use a result of Trotter [43] saying that a graph is a *line perfect* graph, i.e. a root graph of a perfect line graph, if and only if it does not contain any elementary odd cycle of length greater than three. As a consequence, in the same paper, it is obtained that the facets of the matching polytope of a line perfect graph are the stars constraints and the blossom inequalities for induced triangles. It follows that the minimum weighted stars and triangles cover of H is exactly the dual of the maximum matching problem, so we can use one of the many existent primal dual algorithm for the maximum matching to obtain a solution of the weighted stars and triangle edge cover of H, and the solution of this problem will have a computational cost of $O(match(|V(H)|))$.

However, even for an integer weight function, the solution could be half integer in the y variables. We can obtain an integer solution in the following way:

- Consider the actual edge cover (y, π) and consider the graph H' induced by the edges $e = uv \in E(H)$ such that $y_u + y_v + \sum_{t \in T: uv \in t} \pi_t = w_{uv}$ and y_u and y_v are half integer.
- H' is not necessarily an induced subgraph of H, but is line perfect by the characterization theorem of Trotter in [43].
- We can solve the unweighted version of the stars and triangle edge cover on H' via the algorithm proposed by Trotter in time $O(\sqrt{|V(H')|}$ $E(H'))$ (which is less then $O(match(|V(H)|)))$. The solution of this problem will give an integer adjustment of the half integer solution of the stars and triangles weighted edge cover on H.

The resulting complexity of computing a minimum weight clique cover of $L(H)$ is then $O(match(|V(H)|)) = O(|V(H)|^2 log|V(H)|) = O(|V(L(H))|^2$ $log|V(L(H))|) = O(|V(G)|^2 log|V(G)|) = O(match(|V(G)|))$ (using the primal-dual algorithm for maximum weight matching by Gabow [19]).

3.7 Reconstructing a MWCC of G from a MWCC of $\tilde{G}$

In this section we want to show that we can build a MWCC of G from a MWCC of $\tilde{G}$. The first issue we have to address is the 'translation' of all the maximal cliques of $\tilde{G}$ (recall that we can always assume that a MWCC gives a positive value only to maximal cliques) into maximal cliques of G. We will see that this will not always be the case. In order to deal with this problem we detail the structure of $\tilde{G}$ and H.

The second issue will be to show that we can give a new weight function to each strip in order to compute a MWCC w.r.t. this function, and this clique cover together with the one obtained from the cover of $\tilde{G}$ will be a MWCC of G.

3.7.1 The structure of $\tilde{G}$ and H

We first show how to build H. Krausz [27] proved the following:

Lemma 3.19. *[27] A graph $J(W, F)$ is the line graph of a multigraph if and only if there exists a family of cliques $\mathcal{Q}$ such that every edge in F is covered by a clique from the family $\mathcal{Q}$, and moreover every vertex in W belongs to exactly two cliques from the family $\mathcal{Q}$.*

In fact, as soon as we are given a family $\mathcal{Q}$ satisfying Lemma 3.19 w.r.t. $\tilde{G}$, we may build H as follows: each clique $K \in \mathcal{Q}$ corresponds to a vertex v_K of H, and two vertices v_{K_1} and v_{K_2} of H are connected by $|K_1 \cap K_2|$ (parallel) edges. In order to build $\mathcal{Q}$, and therefore H, we start from the set of partition cliques defined by $\mathcal{P}'$. Note that each vertex of $\tilde{G}$ belongs to exactly one partition clique, but for each vertex u_2^i from each strip $\tilde{H}_i^3$, as such a vertex belongs to exactly two partition cliques. Also note that each edge of $\tilde{G}$ is covered by a partition clique, but for each edge $u_2^i u_3^i$ from each strip $\tilde{H}_i^2$. Therefore, in order to "complete" $\mathcal{Q}$, we consider, besides the partition cliques, the following set of *completion* cliques of $\tilde{G}$: a clique $\{v\}$ for each vertex v from each strip $\tilde{H}_i^1$ or $\tilde{H}_i^2$; a clique $\{v\}$ for each vertex $v \equiv u_1^i$ from each strip $\tilde{H}_i^2$; a clique $\{v\}$ for each vertex $v \in \{u_1^i, u_3^i\}$ from each strip $\tilde{H}_i^3$; a clique $\{u_2^i, u_3^i\}$ from each strip $\tilde{H}_i^2$. It is easy to see that the union of the partition and the completion cliques satisfies Lemma 3.19. The next remark summarizes the structure of H.

Remark 4. *Suppose that $\mathcal{P} = \{P_1, \ldots, P_r\}$ and let H be the multigraph such that $L(H) = \tilde{G}$. Then H is composed by: a set of vertices $\{x_1, \ldots, x_r\}$, each x_i corresponding to the class P_i of $\mathcal{P}$.*

- *For each strip H_i such that we use $\tilde{H}_i^3$ instead of H_i in the composition and such that $A_1^i \in P_j$ and $A_2^i \in P_\ell$ we have:*
  - *An edge $x_j x_\ell$ (this edge corresponds to the vertex u_2^i of T_3^i)*
  - *Vertices z_j^i and z_ℓ^i*
  - *Edges $z_j^i x_j$ and $z_\ell^i x_\ell$ (the edges $z_j^i x_j$ and $z_\ell^i x_\ell$ correspond to the vertices u_1^i and u_3^i of T_3^i, respectively)*

- *For each strip H_i such that we use $\tilde{H}_i^2$ instead of H_i in the composition and such that $A_1^i \in P_j$ and $A_2^i \in P_\ell$ we have:*
  - *A vertex $y_{j\ell}^i$*
  - *A vertex z_j^i*
  - *Edges $y_{j\ell}^i x_j$ and $y_{j\ell}^i x_\ell$ (the edges $y_{j\ell}^i x_j$ and $y_{j\ell}^i x_\ell$ correspond to the vertices u_2^i and u_3^i of T_2^i, respectively)*
  - *An edge $z_j^i x_j$ (this edge corresponds to the vertex u_1^i of T_2^i)*
- *for each strip H_i such that we use $\tilde{H}_i^0$ instead of H_i in the composition and such that $A_1^i \in P_j$ we have:*
  - *A vertex z_j^i*
  - *An edge $z_j^i x_j$ (this edge corresponds to the vertex c^i of T_0^i)*
- *For each strip H_i such that we use $\tilde{H}_i^1$ instead of H_i in the composition and such that $A_1^i \in P_j$ and $A_2^i \in P_\ell$ we have:*
  - *Vertices z_j^i and z_ℓ^i*
  - *Edges $z_j^i x_j$ and $z_\ell^i x_\ell$ (the edges $z_j^i x_j$ and $z_\ell^i x_\ell$ correspond to the vertices u_1^i and u_2^i of T_1^i, respectively)*

We now analyze the maximal cliques of $\tilde{G}$. We already pointed out that [43] *each maximal clique of $\tilde{G}$ corresponds to either a multistar of H or to a multitriangle of H.*

We start with multistars of H. By construction, each multistar of H corresponds to either a partition or a completion clique of $\tilde{G}$. It is easy to see that each partition clique is maximal. A completion clique is maximal if it either coincides with some partition clique, or it is a clique $\{u_2^i, u_3^i\}$ from some strip $\tilde{H}_i^2$. In particular, if the extremities of $\tilde{H}_i^2$ belong to the classes P_j and $P_\ell \in \mathcal{P}'$, $\{u_2^i, u_3^i\}$ is maximal if and only if there is no strip $\tilde{H}_a^3$ whose extremities are in the same classes P_j and P_ℓ, as otherwise $\{u_2^i, u_3^i, u_2^a\}$ would be a larger clique.

We now move to multitriangles of H. Trivially, each multitriangle of H induces a maximal clique of $\tilde{G}$. By construction, the multitriangles of H delve into two classes. A first class are those induced by vertices x_j, x_ℓ, x_k

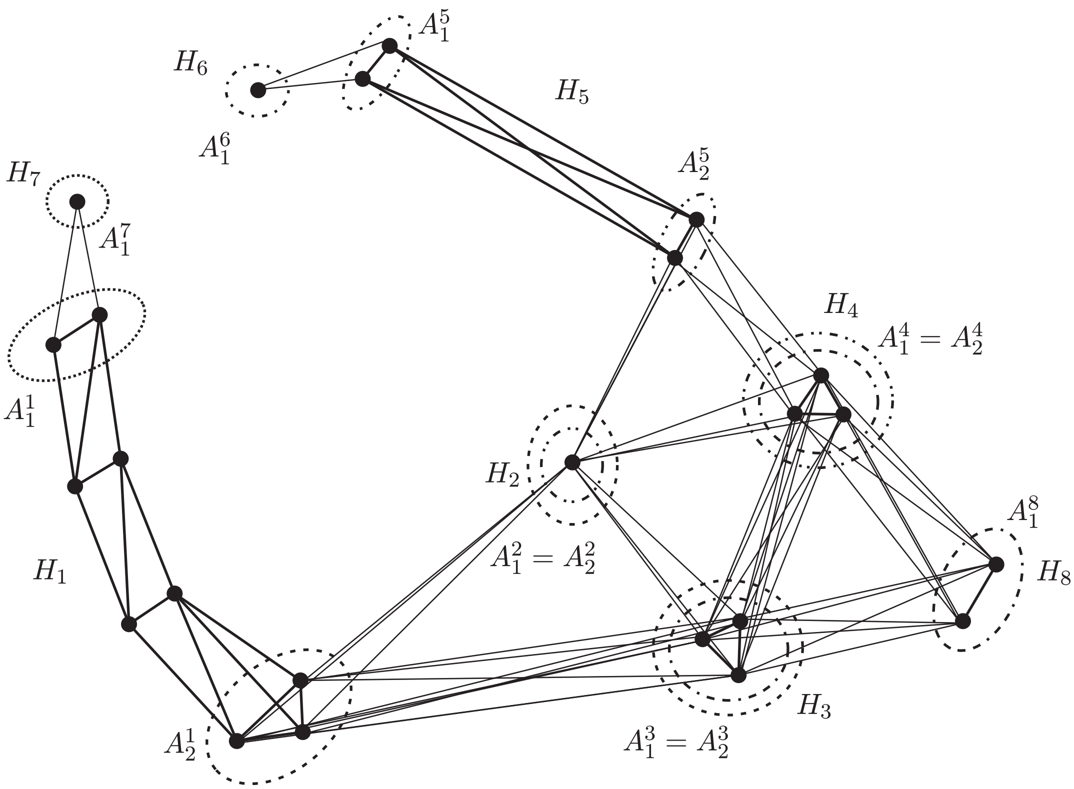


Figure 3.2: A graph G, composition of the 2-strips $H_1, \ldots, H_5$ and the 1-strips H_6, H_7, H_8. Partition $\mathcal{P}$ is given by
$\{\{A_1^1, A_1^7\}, \{A_1^5, A_1^6\}, \{A_2^1, A_2^2, A_2^3\}, \{A_1^2, A_2^5, A_1^4\}, \{A_1^3, A_2^4, A_1^8\}\}$.

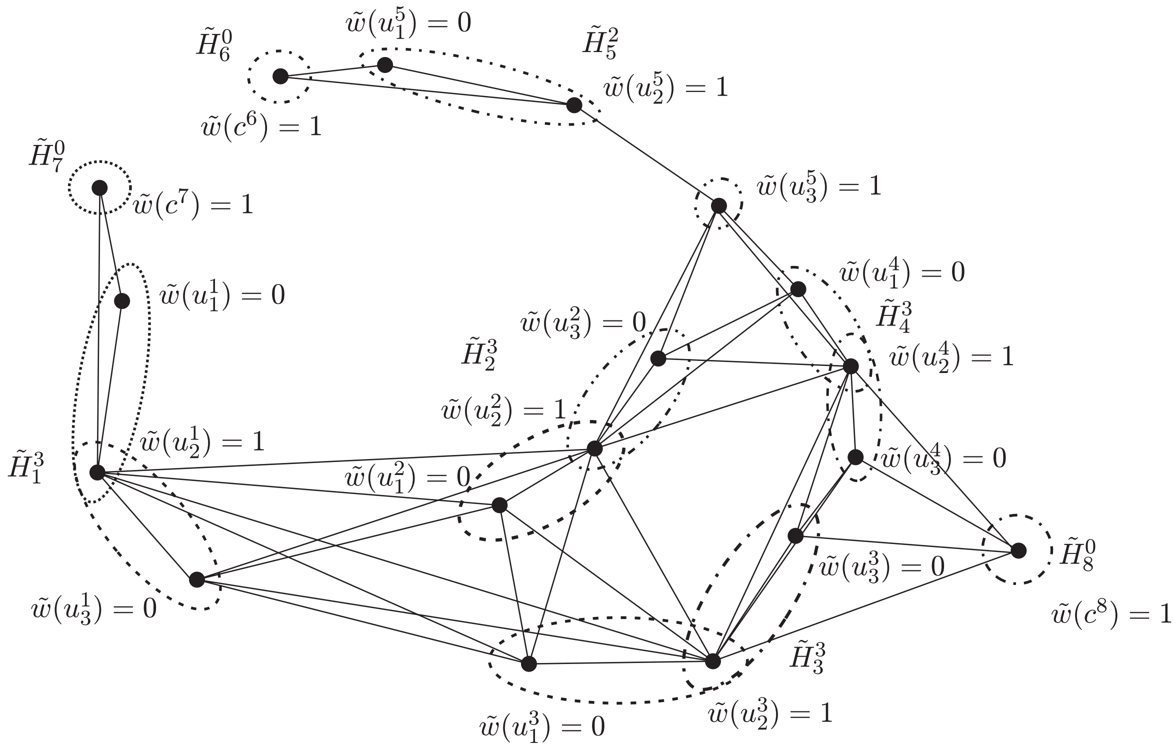


Figure 3.3: The weighted graph $\tilde{G}$, corresponding to graph G in Figure 3.2, and the weight function w such that $w(v) = 1$ for every vertex v of G.

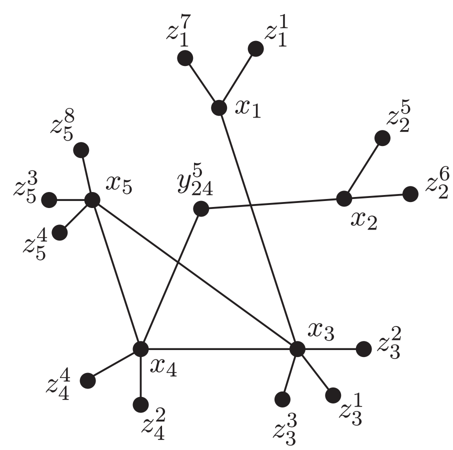


Figure 3.4: The root graph H of the line graph $\tilde{G}$ in Figure 3.3.

such that: there exist $\tilde{H}^3_a, \tilde{H}^3_b, \tilde{H}^3_c$ with the extremities of $\tilde{H}^3_a$ in P_j and P_ℓ, the extremities of $\tilde{H}^3_b$ in P_ℓ and P_k, the extremities of $\tilde{H}^3_c$ in P_k and P_j. By construction, each of the strip H_a, H_b, H_c is either an even strip or an even-odd strip $(G^i, \mathcal{A}^i)$, for $i \in \{a, b, c\}$. Let a_1, a_2 be the endpoints of an even A^a_1–A^a_2 path, and define b_1, b_2 and c_1, c_2 analogously. Then these three paths along with the edges a_2b_1, b_2c_2 and a_1c_1 induce an odd hole, unless the three paths have length zero, i.e, $a_1 = a_2$, $b_1 = b_2$ and $c_1 = c_2$. That is the case when G is a perfect graph. If $a_1 = a_2$, $b_1 = b_2$ and $c_1 = c_2$ then $A^a_1 \cap A^a_2 \neq \emptyset$, $A^b_1 \cap A^b_2 \neq \emptyset$ and $A^c_1 \cap A^c_2 \neq \emptyset$ and by construction $(A^a_1 \cap A^a_2) \cup (A^b_1 \cap A^b_2) \cup (A^c_1 \cap A^c_2)$ is a clique. Moreover, H_a, H_b and H_c are even-short strips.

A second class of multitriangle of H are those induced by vertices $x_j, x_\ell, y^i_{j\ell}$ such that: there exist $\tilde{H}^3_a$ and $\tilde{H}^2_i$ with one of the two extremities in a class P_j and the other in a class P_ℓ. Note that such a multitriangle "arises" from some non-maximal completion cliques of $\tilde{G}$, see above. For these multitriangles we can show the following claim.

Claim 3.20. *Suppose that in H we have a (multi)triangle $y^i_{j\ell}x_jx_\ell$, then there is an odd or even-odd 2-strip $(G^i, \mathcal{A}^i)$ with a vertex x in G complete to both extremities. Moreover H_i is an odd-short strip.*

Proof. As we have in H the edge x_jx_ℓ, there must be in $\tilde{G}$ a corresponding $\tilde{H}^3_k$ gadget with vertex set $\{u^k_1, u^k_2, u^k_3\}$, and in G a corresponding even or even-odd strip $(G^k, \mathcal{A}^k)$ with one of the two extremities in P_j and the other in P_ℓ. As we have the two edges $y^i_{j\ell}x_j$ and $y^i_{j\ell}x_\ell$ there must be in $\tilde{G}$ a corresponding $\tilde{H}^2_i$ gadget with vertex set $\{u^i_1, u^i_2, u^i_3\}$, and in G a corresponding odd or even-odd strip $(G^i, \mathcal{A}^i)$ with one of the two extremities in P_j and the other in P_ℓ. Moreover, we observe that the triangle $y^i_{j\ell}x_jx_\ell$ in H corresponds to the triangle $u^k_2u^i_2u^i_3$ in $\tilde{G}$. The strip $(G^i, \mathcal{A}^i)$ is odd or even-odd, thus it has at least an A^i_1-A^i_2 odd path, while $(G^k, \mathcal{A}^k)$ is even or even-odd, thus it has at least an A^k_1-A^k_2 even path. Then, in order to avoid odd-holes, $A^k_1 \cap A^k_2 \neq \emptyset$ and every odd path in G^i should be of length one. Thus $(G^k, \mathcal{A}^k)$ is an even-short strip, the intersection $A^k_1 \cap A^k_2$ is nonempty and complete to $A^i_1 \cup A^i_2$, and $(G^i, \mathcal{A}^i)$ is an odd-short strip. □

3.7.2 From a MWCC of $\tilde{G}$ to a MWCC of G

A dual solution to the maximum weighted matching problem on H will give a weight to each multistar and multitriangle of H (and w.l.o.g. we assume that this weight is non-zero only for multistars and multitriangles of H correspoding to maximal clique of $\tilde{G}$). This solution is trivially a MWCC of $\tilde{G}$. We are left to show how to "translate" a MWCC $\tilde{y}$ of $\tilde{G}$, w.r.t. the weight $\tilde{w}$, into a MWCC y of G, w.r.t. the weight w.

First, we would like to associate to each maximal clique of $\tilde{G}$ a maximal clique of G. However, there is a catch: the maximal cliques of $\tilde{G}$ arising from the completion cliques might not correspond to any clique of G. We will address this issue later; while we now assume that:

(*) *no strip H_i has been replaced by the strip $\tilde{H}_i^2 = (T_2^i, \tilde{\mathcal{A}}_2^i)$, i.e., each maximal cliques of $\tilde{G}$ corresponds to a maximal clique of G.*

In particular, we will show which clique of G to associate to a maximal clique $\tilde{K}$ of $\tilde{G}$ corresponding to either a multistar centered at some vertex x_j of H, or to a multitriangle $x_j x_\ell x_k$ of H. In the former case, we will translate $\tilde{K}$ into the partition-clique $K = \bigcup_{A_d^i \in P_j} A_d^i$, and set $y(K) = \tilde{y}(\tilde{K})$. In the latter case, following the discussion in the previous section, we will translate $\tilde{K}$ into the clique induced by $K = \bigcup_{d \in I_{j\ell k}} (A_1^d \cap A_2^d)$, where $I_{j\ell k}$ is the set of indices d of even-short 2-strips $H_d = (G^d, \mathcal{A}^d)$ in the decomposition having their two extremities in two different sets in $\{P_j, P_\ell, P_k\}$, and again set $y(K) = \tilde{y}(\tilde{K})$.

Now we want to show that we can "extend" y (we refer to y in the following as a *partial cover*) into a MWCC of G, w.r.t. the weight w. As we show in the following, we will be able to cover the "residual" weight of the vertices in each strip $H_i = (G^i, \mathcal{A}^i)$, building upon a suitable weighted clique cover of G^i of value at most δ^i.

We first deal with 1-strips. Let $H_i = (G^i, \mathcal{A}^i)$ be a 1-strip. H_i has been replaced by the strip $\tilde{H}_i^0 = (T_0^i, \tilde{\mathcal{A}}_0^i)$, where the graph T_0^i consists on a single vertex c^i, and $\tilde{\mathcal{A}}_0^i = \{\{c^i\}\}$. It follows from the discussion in Section 3.7.1 that there is only one clique in the support of $\tilde{y}$ covering $\tilde{\mathcal{A}}_0^i = \{\{c^i\}\}$, and that clique corresponds in G to the partition clique from the class of $\mathcal{P}$ which A_1^i belongs to. Therefore, each vertex in A_1^i is covered by a single clique in the support of y, with weight at least $\tilde{w}(c^i) = \alpha_w(G^i) - \delta_1^i$, where $\delta_1^i = \alpha_w(G^i \setminus A_1^i)$. Then we can "extend" y into a MWCC of G, w.r.t. w, because of the following lemma:

Lemma 3.21. *Let $H_i = (G^i, \mathcal{A}^i)$ be a 1-strip. Let $w^i : V(G^i) \mapsto \mathbb{R}^+$ be defined as follows:*

$w^i(v) = w(v)$ *for* $v \in V(G^i) \setminus A_1^i$;

$w^i(v) = \max\{0, w(v) - b\}$ *for* $v \in A_1^i$

with $b \geq \alpha_w(G^i) - \delta_1^i$. *Then* $\alpha_{w^i}(G^i) = \delta_1^i$.

Proof. Let S be a stable set of $G^i \setminus A_1^i$. Since $w^i(v) = w(v)$ for $v \in V(G^i) \setminus A_1^i$, it follows that $w^i(S) = w(S) \leq \alpha_w(G^i \setminus A_1^i) = \delta_1^i$. Now consider a stable set S of G^i containing one vertex $v \in A_1^i$. If $w^i(v) \leq 0$, then $w^i(S) = w(S \setminus v) + w^i(v) \leq w(S \setminus v) \leq \alpha_w(G^i \setminus A_1^i) = \delta_1^i$; if $w^i(v) > 0$, then $w^i(S) = w(S) - b \leq w(S) - \alpha_w(G^i) + \delta_1^i \leq \delta_1^i$. □

We now move to 2-strips that have been replaced by $\tilde{H}^0$, i.e. strips with both extremities in the same class of $\mathcal{P}$. Let $H_i = (G^i, \mathcal{A}^i)$ be such a strip. H_i has been replaced by the strip $\tilde{H}_i^0 = (T_0^i, \tilde{\mathcal{A}}_0^i)$, where the graph T_0^i consists on a single vertex c^i, and $\tilde{\mathcal{A}}_0^i = \{\{c^i\}\}$. It follows from the discussion in Section 3.7.1 that there is only one clique in the support of $\tilde{y}$ covering $\tilde{\mathcal{A}}_0^i = \{\{c^i\}\}$, and that clique corresponds in G to the partition clique from the class of $\mathcal{P}$ which A_1^i and A_2^i belong to. Therefore, each vertex in $A_1^i \cup A_2^i$ is covered by a single clique in the support of y, with weight at least $\tilde{w}(c^i) = \max\{\alpha_w(G^i \setminus A_1^i), \alpha_w(G^i \setminus A_2^i), \alpha_w(G^i \setminus (A_1^i \triangle A_2^i))\} - \delta_1^i$, where $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$. Then we can "extend" y into a MWCC of G, w.r.t. w, because of the following lemma:

Lemma 3.22. *Let $H_i = (G^i, \mathcal{A}^i)$ be a 2-strip. Let $G_=^i$ be the graph obtained from G^i by adding the edges between A_1^i and A_2^i. Let $w^i : V(G^i) \mapsto \mathbb{R}^+$ be defined as follows:*

$w^i(v) = w(v)$ *for* $v \in V(G^i) \setminus (A_1^i \cup A_2^i)$;

$w^i(v) = \max\{0, w(v) - b\}$ *for* $v \in A_1^i \cup A_2^i$

with $b \geq \max\{\alpha_w(G^i \setminus A_1^i), \alpha_w(G^i \setminus A_2^i), \alpha_w(G^i \setminus (A_1^i \triangle A_2^i))\} - \delta_1^i$. *Then* $\alpha_{w^i}(G^i) = \delta_1^i$. *Moreover, if* $G_=^i$ *is perfect, any* MWCC *of* $G_=^i$ *w.r.t.* w^i *does not assign strictly positive weight to the clique* $A_1^i \cup A_2^i$.

Proof. First note that $\alpha_w(G_=^i) = \max\{\alpha_w(G^i \setminus A_1^i), \alpha_w(G^i \setminus A_2^i), \alpha_w(G^i \setminus (A_1^i \triangle A_2^i))\}$. Now, let S be a maximum stable set of $G^i \setminus (A_1^i \cup A_2^i)$ w.r.t. w. Since $w^i(v) = w(v)$ for $v \in V(G^i) \setminus (A_1^i \cup A_2^i)$, then $w^i(S) = w(S) = \alpha_w(G^i \setminus (A_1^i \cup A_2^i)) = \delta_1^i$. On the other hand and by the same reason, any stable set S of $G_=^i$ such that $w^i(S) > \delta_1^i$ should contain a vertex $v \in A_1^i \cup A_2^i$, such that $w^i(v) > 0$, i.e., $w^i(v) = w(v) - b$. Since it is a clique of $G_=^i$, there is at most one such vertex. So, $w^i(S) = w(S) - b \leq w(S) - \alpha_w(G_=^i) + \delta_1^i \leq \delta_1^i$. Then $\alpha_{w^i}(G_=^i) = \delta_1^i$. If $G_=^i$ is perfect, any MWCC of $G_=^i$ with respect to w^i should have weight δ_1^i. In particular, every clique with strictly positive weight must intersect any MWSS of $G^i \setminus (A_1^i \cup A_2^i)$. So, in any MWCC of $G_=^i$, the clique $A_1^i \cup A_2^i$ has weight zero. □

Note that the last sentence of the previous lemma implies that, if $G_=^i$ is perfect and there are no two vertices $v_1 \in A_1^i$ and $v_2 \in A_2^i$ having a common neighbor in $V(G^i) \setminus (A_1^i \cup A_2^i)$, then any MWCC of $G_=^i$ w.r.t. w^i is in fact a MWCC of G^i w.r.t. w^i. Otherwise, we must be able to compute a MWCC of $G_=^i$ in order to reconstruct a clique cover of G from a clique cover of $\tilde{G}$. This is why we require in Theorem 3.5 that a MWCC of $G_=^i$ can be computed in time $O(p_i(|V(G^i)|))$ in this case.

We now move to 2-strips that have been replaced by $\tilde{H}^1$. Let $H_i = (G^i, \mathcal{A}^i)$ be such a strip. H_i has been replaced by the strip $\tilde{H}_i^1 = (T_1^i, \tilde{\mathcal{A}}_1^i)$, where $V(T_1^i) = \{u_1^i, u_2^i\}$, $E(T_1^i) = \emptyset$, $\tilde{\mathcal{A}}_1^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i\}$, $\tilde{A}_2^i = \{u_2^i\}$. It follows from the discussion in Section 3.7.1 that there is only one clique in the support of $\tilde{y}$ covering $\tilde{A}_1^i = \{u_1^i\}$, and that clique corresponds in G to the partition clique from the class of $\mathcal{P}$ which A_1^i belongs to, and the same holds w.r.t. to u_2^i and A_2^i. Therefore, each vertex in A_1^i is covered by a single clique in the support of y, with weight at least $\tilde{w}(u_1^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, and each vertex in A_2^i is covered by a single clique in the support of y, with weight at least $\tilde{w}(u_2^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$, where $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$. Recall that $\alpha_w(G^i \setminus A_1^i) + \alpha_w(G^i \setminus A_2^i) \geq \alpha_w(G^i) + \delta_1^i$, as we replaced H_i

with $\tilde{H}_i^1$. Then we can "extend" y into a MWCC of G, w.r.t. w, because of the following lemma:

Lemma 3.23. *Let $H_i = (G^i, \mathcal{A}^i)$ be a 2-strip. Let $w^i : V(G^i) \mapsto \mathbb{R}^+$ be defined as follows:*

$w^i(v) = w(v)$ *for* $v \in V(G^i) \setminus (A_1^i \cup A_2^i)$*;*

$w^i(v) = \max\{0, w(v) - b_1\}$ *for* $v \in A_1^i \setminus A_2^i$*;*

$w^i(v) = \max\{0, w(v) - b_2\}$ *for* $v \in A_2^i \setminus A_1^i$*;*

$w^i(v) = \max\{0, w(v) - b_1 - b_2\}$ *for* $v \in A_1^i \cap A_2^i$*.*

with b_1, b_2 such that $b_1 \geq \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, $b_2 \geq \alpha_w(G^i \setminus A_1^i) - \delta_1^i$, and $b_1 + b_2 \geq \alpha_w(G^i) - \delta_1^i$. Then $\alpha_{w^i}(G^i) = \delta_1^i$.

Proof. On one hand, let S be a MWSS of $G^i \setminus (A_1^i \cup A_2^i)$ w.r.t. w. Since $w^i(v) = w(v)$ for $v \in V(G^i) \setminus (A_1^i \cup A_2^i)$, then $w^i(S) = w(S) = \alpha_w(G^i \setminus (A_1^i \cup A_2^i)) = \delta_1^i$. On the other hand and by the same reason, any stable set S such that $w^i(S) > \delta_1^i$ should contain a vertex $v \in A_1^i \cup A_2^i$ such that $w^i(v) > 0$. In fact, w.l.o.g., we can assume that every vertex in S has strictly positive weight. Now, we have four cases to consider: S contains a vertex v of A_1^i and no vertex of A_2^i; S contains a vertex v of A_2^i and no vertex of A_1^i; S contains a vertex v of $A_1^i \cap A_2^i$; or S contains a vertex v of $A_1^i \setminus A_2^i$ and a vertex v' of $A_2^i \setminus A_1^i$. In the first case, $w^i(v) = w(v) - b_1$ and so $w^i(S) = w(S) - b_1 \leq w(S) - \alpha_w(G^i \setminus A_2^i) + \delta_1^i \leq \delta_1^i$. The second case is symmetric. In the last two cases, $w^i(S) = w(S) - b_1 - b_2 \leq w(S) - \alpha_w(G^i) + \delta_1^i \leq \delta_1^i$. □

We now move to 2-strips that have been replaced by $\tilde{H}_i^3$ but are not even-short, i.e. $A_1^i \cap A_2^i = \emptyset$. Let $H_i = (G^i, \mathcal{A}^i)$ be such a strip. H_i has been replaced by the strip $\tilde{H}_i^3 = (T_3^i, \tilde{\mathcal{A}}_3^i)$, with $V(T_3^i) = \{u_1^i, u_2^i, u_3^i\}$, $E(T_3^i) = \{u_1^i u_2^i, u_2^i u_3^i\}$, $\tilde{\mathcal{A}}_3^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i, u_2^i\}$, $\tilde{A}_2^i = \{u_2^i, u_3^i\}$. It follows from the discussion in Section 3.7.1 that there is only one clique in the support of $\tilde{y}$ covering u_1^i (resp. u_3^i), and that clique corresponds in G to the partition clique from the class of $\mathcal{P}$ which A_1^i (resp. A_2^i) belongs to. Therefore, as A_1^i and A_2^i do not intersect, each vertex in A_1^i is covered by a single clique in the support of y, with weight at least $\tilde{w}(u_1^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, where $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$; analogously, each vertex in A_2^i is covered by a single clique in the support of y, with weight at least $\tilde{w}(u_3^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$. Note also that $\tilde{w}(u_1^i) + \tilde{w}(u_3^i) \geq \tilde{w}(u_2^i) = \alpha_w(G^i) - \delta_1^i$, as the only maximal cliques of $\tilde{G}$ covering u_2^i contains either u_1^i or u_3^i; therefore, $\alpha_w(G^i \setminus A_1^i) + \alpha_w(G^i \setminus A_2^i) \geq \alpha_w(G^i) + \delta_1^i$. Then we can "extend" y into a MWCC of G, w.r.t. w, because of Lemma 3.23 again.

We now move to 2-strips that have been replaced by $\tilde{H}_i^3$ and are even-short, i.e. $A_1^i \cap A_2^i \neq \emptyset$. Note that such strips might be involved in some multitriangles in the root graph of $\tilde{G}$.

Let $H_i = (G^i, \mathcal{A}^i)$ be such a strip. H_i has been replaced by the strip $\tilde{H}_i^3 = (T_3^i, \tilde{\mathcal{A}}_3^i)$, with $V(T_3^i) = \{u_1^i, u_2^i, u_3^i\}$, $E(T_3^i) = \{u_1^i u_2^i, u_2^i u_3^i\}$, $\tilde{\mathcal{A}}_3^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i, u_2^i\}$, $\tilde{A}_2^i = \{u_2^i, u_3^i\}$. It follows from the discussion in Section 3.7.1 that there is only one clique in the support of $\tilde{y}$ covering u_1^i (resp. u_3^i), and that clique corresponds in G to the partition clique from the class of $\mathcal{P}$ which A_1^i (resp. A_2^i) belongs to. Therefore, each vertex in $A_1^i \setminus A_2^i$ is covered by a single clique in the support of y, with weight at least $\tilde{w}(u_1^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, where $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$; analogously, each vertex in $A_2^i \setminus A_1^i$ is covered by a single clique in the support of y, with weight at least $\tilde{w}(u_3^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$. As for the vertices in $A_1^i \cap A_2^i$, they might be also covered, with weight $a \geq 0$, by some clique $K = \bigcup_{d \in I_{j\ell k}} (A_1^d \cap A_2^d)$, with $i \in I_{j\ell k}$ (recall that $I_{j\ell k}$ is the set of indices d of even-short 2-strips $H_d = (G^d, \mathcal{A}^d)$ in the decomposition having their two extremities in two different sets in $\{P_j, P_\ell, P_k\}$). Note that it follows that $\tilde{w}(u_1^i) + \tilde{w}(u_3^i) + a \geq \tilde{w}(u_2^i) = \alpha_w(G^i) - \delta_1^i$, as the only maximal cliques of $\tilde{G}$ covering u_2^i contains either u_1^i or u_3^i; therefore, $\alpha_w(G^i \setminus A_1^i) + \alpha_w(G^i \setminus A_2^i) + a \geq \alpha_w(G^i) + \delta_1^i$. Then we can "extend" y into a MWCC of G, w.r.t. w, because of the following lemma:

Lemma 3.24. *Let $H_i = (G^i, \mathcal{A}^i)$ be an even-short 2-strip such that G^i is perfect. Let $w^i : V(G^i) \mapsto \mathbb{R}^+$ be defined as follows:*

$w^i(v) = w(v)$ *for* $v \in V(G^i) \setminus (A_1^i \cup A_2^i)$;

$w^i(v) = \max\{0, w(v) - b_1\}$ *for* $v \in A_1^i \setminus A_2^i$;

$w^i(v) = \max\{0, w(v) - b_2\}$ *for* $v \in A_2^i \setminus A_1^i$;

$w^i(v) = \max\{0, w(v) - b_1 - b_2 - a\}$ *for* $v \in A_1^i \cap A_2^i$.

with b_1, b_2, a *be such that* $b_1 \geq \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, $b_2 \geq \alpha_w(G^i \setminus A_1^i) - \delta_1^i$, *and* $a + b_1 + b_2 \geq \alpha_w(G^i) - \delta_1^i$. *Then* $\alpha_{w^i}(G^i) = \delta_1^i$.

Proof. On one hand, let S be a MWSS of $G^i \setminus (A_1^i \cup A_2^i)$ w.r.t. w. Since $w^i(v) = w(v)$ for $v \in V(G^i) \setminus (A_1^i \cup A_2^i)$, then $w^i(S) = w(S) = \alpha_w(G^i \setminus (A_1^i \cup A_2^i)) = \delta_1^i$. On the other hand and by the same reason, any stable set S such that $w^i(S) > \delta_1^i$ should contain a vertex $v \in A_1^i \cup A_2^i$ such that $w^i(v) > 0$. In fact, w.l.o.g., we can assume that every vertex in S has strictly positive weight. Now, we have four cases to consider: S contains a vertex v of A_1^i and no vertex of A_2^i; S contains a vertex v of A_2^i and no vertex of A_1^i; S contains a vertex v of $A_1^i \cap A_2^i$; or S contains a vertex v of $A_1^i \setminus A_2^i$ and a vertex v' of $A_2^i \setminus A_1^i$. In the first case, $w^i(v) = w(v) - b_1$ and so $w^i(S) = w(S) - b_1 \leq w(S) - \alpha_w(G^i \setminus A_2^i) + \delta_1^i \leq \delta_1^i$. The second case is symmetric. In the third case, $w^i(v) = w(v) - b_1 - b_2 - a$, and so $w^i(S) = w(S) - b_1 - b_2 - a \leq w(S) - \alpha_w(G^i) + \delta_1^i \leq \delta_1^i$. In the last case, $w^i(v) = w(v) - b_1$ and $w^i(v') = w(v') - b_2$, and so $w^i(S) = w(S) - b_1 - b_2 \leq w(S) - \alpha_w(G^i \setminus A_2^i) - \alpha_w(G^i \setminus A_1^i) + 2\delta_1^i$. Note that, since H_i is an even-short strip, the strip $H_i' = (G^i \setminus (A_1^i \cap A_2^i), \{A_1^i \setminus A_2^i, A_2^i \setminus A_1^i\})$ is either non-connected or odd, so by Lemma 3.17, $\alpha_w(G^i \setminus A_1^i) + \alpha_w(G^i \setminus A_2^i) \geq \alpha_w(G^i \setminus (A_1^i \cap A_2^i)) + \delta_1^i$. Thus $w^i(S) \leq w(S) - \alpha_w(G^i \setminus (A_1^i \cap A_2^i)) + \delta_1^i \leq \delta_1^i$. □

3.7.3 When some strip is replaced by the strip $\tilde{H}_i^2 = (T_2^i, \tilde{\mathcal{A}}_2^i)$

We finally deal with the case where the hypothesis (*) does not hold, i.e. some strip H_i has been replaced by the strip $\tilde{H}_i^2 = (T_2^i, \tilde{\mathcal{A}}_2^i)$.

Let $H_i = (G^i, \mathcal{A}^i)$ be such a strip. H_i has been replaced by the strip $\tilde{H}_i^2 = (T_2^i, \tilde{\mathcal{A}}_2^i)$, with $V(T_2^i) = \{u_1^i, u_2^i, u_3^i\}$, $E(T_2^i) = \{u_1^i u_2^i, u_2^i u_3^i\}$, $\tilde{\mathcal{A}}_2^i = \{\tilde{A}_1^i, \tilde{A}_2^i\}$ and $\tilde{A}_1^i = \{u_1^i, u_2^i\}$, $\tilde{A}_2^i = \{u_3^i\}$. We also set: $\tilde{w}(u_1^i) = \alpha_w(G^i) - \alpha_w(G^i \setminus A_1^i)$, $\tilde{w}(u_2^i) = \alpha_w(G^i \setminus A_2^i) - \delta_1^i$, $\tilde{w}(u_3^i) = \alpha_w(G^i \setminus A_1^i) - \delta_1^i$.

Recall that, in this case, there might be some maximal cliques of $\tilde{G}$ that do not correspond to any maximal clique of G. We will show how to define a weight function on the vertices of the strip H_i as to get a cover with the same value which includes only cliques.

Recall that these "fake" cliques correspond in the root graph H to some star centered at $y^i_{j\ell}$ and to some multitriangle $y^i_{j\ell}x_jx_\ell$. We first deal with the case where in H there is a multitriangle $y^i_{j\ell}x_jx_\ell$. In this case, the 2-strip $(G^i, \mathcal{A}^i)$ is odd-short, and there is at least one vertex x that is complete to $A^i_1 \cup A^i_2$, and belongs to some strip $(G^a, \mathcal{A}^a)$ whose extremities are in the same class as $(G^i, \mathcal{A}^i)$. Without loss of generality, we assume that x is unique (in fact, if there are more vertices, then they form a clique of G). We prove the following lemma, which essentially shows that, if our cover of $\tilde{G}$ has assigned a weight $a > 0$ to the triangle $y^i_{j\ell}x_jx_\ell$, then we can discard this triangle and ask for a MWCC of value $\delta^i_1 + a$ in the graph induced by G^i and x, in such a way that x is covered by a quantity greater or equal to a.

Lemma 3.25. *Let $H_i = (G^i, \mathcal{A}^i)$ be a 2-strip. Let $G^i_\bullet$ be the graph obtained from G^i by adding a new vertex x complete to both A^i_1 and A^i_2. Let $w^i : V(G^i) \mapsto \mathbb{R}^+$ be defined as follows:*

$w^i(v) = w(v)$ *for* $v \in V(G^i) \setminus (A^i_1 \cup A^i_2)$

$w^i(v) = \max\{0, w(v) - b_1\}$ *for* $v \in A^i_1 \setminus A^i_2$,

$w^i(v) = \max\{0, w(v) - b_2\}$ *for* $v \in A^i_2 \setminus A^i_1$;

$w^i(v) = \max\{0, w(v) - b_1 - b_2\}$ *for* $v \in A^i_1 \cap A^i_2$.

with b_1, b_2, a such that $b_1 \geq \alpha_w(G^i) - \alpha_w(G^i \setminus A^i_1)$, $a + b_1 \geq \alpha_w(G^i \setminus A^i_2) - \delta^i_1$, $a + b_2 \geq \alpha_w(G^i \setminus A^i_1) - \delta^i_1$. Then $\alpha_{w^i}(G^i_\bullet) = \delta^i_1 + a$. In particular, $\alpha_{w^i}(G^i) \leq \delta^i_1 + a$.

Proof. On one hand, let S be a MWSS of $G^i \setminus (A^i_1 \cup A^i_2)$ w.r.t. w. Then $S \cup \{x\}$ is a stable set of $G^i_\bullet$. Since $w^i(v) = w(v)$ for $v \in V(G^i) \setminus (A^i_1 \cup A^i_2)$, then $w^i(S \cup \{x\}) = w(S) + w^i(x) = \alpha_w(G^i \setminus (A^i_1 \cup A^i_2)) + a = \delta^i_1 + a$. In fact, since x is complete to $A^i_1 \cup A^i_2$ in $G^i_\bullet$, any stable set of $G^i_\bullet$ containing x should be composed by x and a stable set of $G^i \setminus (A^i_1 \cup A^i_2)$, and will have weight w^i at most $\delta^i_1 + a$. So, any stable set S such that $w^i(S) > \delta^i_1 + a$ should contain a vertex $v \in A^i_1 \cup A^i_2$ such that $w^i(v) > 0$. In fact, w.l.o.g., we can assume that every vertex in S has strictly positive weight. Now, we have four cases to consider: S contains a vertex v of A_1^i and no vertex of A_2^i; S contains a vertex v of A_2^i and no vertex of A_1^i; S contains a vertex v of $A_1^i \setminus A_2^i$ and a vertex v' of $A_2^i \setminus A_1^i$; or S contains a vertex v of $A_1^i \cap A_2^i$. In the first case, $w^i(v) = w(v) - b_1$ and so $w^i(S) = w(S) - b_1 \leq w(S) - \alpha_w(G^i \setminus A_2^i) + \delta_1^i + a \leq \delta_1^i + a$. The second case is symmetric. In the third case, $w^i(v) = w(v) - b_1$ and $w^i(v') = w(v') - b_2$, and in the last case $w^i(v) = w(v) - b_1 - b_2$. So, in both cases, $w^i(S) = w(S) - b_1 - b_2$. By adding the first two required inequalities, it follows that $a + b_1 + b_2 \geq \alpha_w(G^i) - \delta_1^i$, so $w^i(S) \leq w(S) - \alpha_w(G^i) + \delta_1^i + a \leq \delta_1^i + a$. □

We underline that the last sentence of Lemma 3.25 suggests also how to "translate" the weight a possibly assigned to the star centered in $y_{j\ell}^i$ and, in general, how to deal with the strips that have been replaced by $\tilde{H}^2$.

3.8 Conclusions

From the collection of all the results in this chapter we can finally write an algorithm for computing a MWCC in a strip-composed graph G when the decomposition is given.

Clearly under the hypothesis of Theorem 3.5, Algorithm 2 finds a MWCC in polynomial time, with running time $O(\sum_{i=1}^{k} p_i(|V(G^i)|) + match(|V(G)|))$.

Algorithm 2

Require: A graph $G(V,E)$, its strip decomposition $(\mathcal{G},\mathcal{P})$ where $\mathcal{G} = \{(G^j,\mathcal{A}^j), j \in [k]\}$ and a weight function $w : V \to \mathbb{R}^+$.

Ensure: A MWCC for $G(V,E)$ w.r.t. the weight function w.

1: **Replacement Step**:

1.1 For each 1-strip $H_i = (G^i,\mathcal{A}^i)$ compute $\delta_1^i = \tau_w(G^i \setminus A_1^i)$ and $\tau_w(G^i)$; replace H_i with the gadget $\tilde{H}_i^0$ and modify $\mathcal{P}$ accordingly to this replacement.

1.2 For each 2-strip $H_i = (G^i,\mathcal{A}^i)$:

1.2.1 If $\exists P \in \mathcal{P}$ such that $A_1^i, A_2^i \in P$, then compute $\delta_1^i = \tau_w(G^i \setminus (A_1^i \cup A_2^i))$, $\tau_w(G^i \setminus A_1^i)$, $\tau_w(G^i \setminus A_2^i)$ and $\tau_w(G^i \setminus (A_1^i \bigtriangleup A_2^i))$; replace H_i with the gadget $\tilde{H}_i^0$ and modify $\mathcal{P}$ accordingly to this replacement.

1.2.2 Else compute $\tau_w(G^i \setminus A_1^i)$, $\tau_w(G^i \setminus A_2^i)$, $\tau_w(G^i)$ and $\delta_1^i = \tau_w(G^i \setminus (A_1^i \cup A_2^i))$:

- If $\tau_w(G^i \setminus A_1^i) + \tau_w(G^i \setminus A_2^i) = \tau_w(G^i) + \tau_w(G^i \setminus (A_1^i \cup A_2^i))$ replace H_i with the gadget $\tilde{H}_i^1$ and modify $\mathcal{P}$ accordingly to this replacement;
- If $\tau_w(G^i \setminus A_1^i) + \tau_w(G^i \setminus A_2^i) > \tau_w(G^i) + \tau_w(G^i \setminus (A_1^i \cup A_2^i))$ replace H_i with the gadget $\tilde{H}_i^2$ and modify $\mathcal{P}$ accordingly to this replacement;
- If $\tau_w(G^i \setminus A_1^i) + \tau_w(G^i \setminus A_2^i) < \tau_w(G^i) + \tau_w(G^i \setminus (A_1^i \cup A_2^i))$ replace H_i with the gadget $\tilde{H}_i^3$ and modify $\mathcal{P}$ accordingly to this replacement;

2: **Line perfect graph step**: Let $\tilde{G}$ and $\tilde{w}$ be the graph and the weight function obtained from the Replacement Step. $\tilde{G}$ is a perfect and line graph (cfr Remarks 1, 2, 3).

2.1 Compute the root multigraph H of $\tilde{G}$ (cfr Section 3.7.1).

2.2 Compute a MWCC $\tilde{y}$ of $\tilde{G}$ w.r.t to the weight function $\tilde{w}$ (cfr Section 3.6).

3: **Reconstruction step**:

3.1 For each clique with $\tilde{y}(\tilde{K}) \neq 0$ that corresponds to a maximal clique of G (to check this it is sufficient to check from which variable of the stars and triangle edge cover of H comes the clique $\tilde{K}$) set K according to Section 3.7.2 and $y(K) = \tilde{y}(\mathcal{K})$.

3.2 For each clique with $\tilde{y}(\tilde{K}) \neq 0$ that *does not* correspond to a maximal clique of G (to check this it is sufficient to check from which variable of the stars and triangle edge cover of H comes the clique $\tilde{K}$) add a new vertex x complete to both extremities to a suitable odd or even-odd 2-strip and assign to it a weight $w^i(x) = \tilde{y}(\mathcal{K})$ (cfr Section 3.7.3).

3.3 For each strip $H_i = (G^i,\mathcal{A}^i)$ with $i = 1,\dots,k$ compute from $\tilde{y}$ the new weight function w^i according to Lemmas 3.21, 3.22, 3.23 and 3.24.

3.4 For each strip $H_i = (G^i,\mathcal{A}^i)$ with $i = 1,\dots,k$ compute a MWCC y_i of G^i of value δ_1^i w.r.t. the new weight function w^i. If H_i is a 2-strip with both extremities in the same class of the partition and Lemma 3.22 does not apply, compute a MWCC y_i of $G_=^i$ w.r.t. w^i. If H_i is a 2-strip to which we have added a new vertex x in Step 3.2, compute a MWCC y_i of $G_\bullet^i$ w.r.t. w^i.

3.5 For every clique K set $y'(K) = y(K) + \sum_i y_i(K)$.

4: **Return** y'.

Chapter 4

The MWCC problem on strip-composed claw-free perfect graphs

4.1 Introduction

In this chapter we want to apply the result presented in Chapter 3 on the MWCC on strip-composed perfect graphs to the subclass of claw-free perfect graphs, or in other words we want to customize Algorithm 2 to the claw-free case. For the MWCC on claw-free perfect graphs an algorithm has been proposed by Hsu and Nemhauser in 1984 (see [23]); this algorithm mimics the algorithm for the unweighted case proposed by the same authors in [22] and in particular it starts with the computation of a MWSS of the graph G (for a more detailed description of this algorithm see Section 4.3).

Our aim for the class of claw-free perfect graphs was to design an algorithm for the MWCC problem that did not need to compute *any* MWSS. With this aim in mind we have tackled the problem on the subclass of strip-composed claw-free perfect graphs, because in the last decade the structure of quasi-line graphs was deeply investigated, with some results that give a very detailed description and characterization of the strips that, through composition, can be part of a quasi-line graph.

Among all the results on the structure of quasi-line graphs (for a brief outline see Section 4.2) a relevant role in this chapter is played by an algorithmic decomposition theorem by Faenza, Oriolo and Stauffer [17]. The key idea in this chapter is to combine their result with our theorem on the MWCC on strip-composed perfect graphs and obtain from this combination a combinatorial algorithm for the MWCC on strip-composed claw-free perfect graphs. From Theorem 3.5 this algorithm can be polynomial if we can solve in polynomial time the MWCC problem on the strips: thanks to the characterization of the strips given by Faenza, Oriolo and Stauffer we show that this is possible (in Sections 4.5 and 4.6).

4.2 Structure results for quasi-line graphs

We have already underlined in the introductory chapter that the structure of claw-free graphs has been fully investigated in the outstanding series of papers by Chudnovsky and Seymour. One of the paper of this series is specifically dedicated to the subclass of quasi-line graphs (see [10]): the main result of this paper is the following structure theorem.

Theorem 4.1. *[10] Every connected quasi-line trigraph G is either a linear interval join or a thickening of a circular interval trigraph.*

It is not of interest to introduce here what is a trigraph, a thickening or a circular interval trigraph. What we want to notice is that Theorem 4.1 says that either G is in a very well known subclass of trigraphs (i.e. a thickening of a circular interval trigraph), or it is the *composition* of some basic classes of trigraphs (i.e. a linear interval join). Chudnovsky and Seymour in the introduction of the paper explain that a special paper was needed for quasi-line graphs especially for the cases of graphs with $\alpha(G) \leq 3$; in particular every graph with $\alpha(G) = 2$ is claw-free but not necessarly quasi-line. The problem with this kind of result is that, whilst it is very useful in terms of graph structure, because it gives a very detailed description of all the 'building blocks' of the linear interval join, it cannot be easily exploited from an algorithmic point of view, or in other words it is not clear *if* and *how* we can find those building blocks in polynomial time. In fact there is no algorithm for a combinatorial optimization problem using this result

(Chudnovsky herself in a paper with Ovetsky where they give bounds for the $\chi(G)$ of a quasi-line graph and give an algorithm to find an approximate coloring (see [5]), does not use this theorem).

In [34] the authors propose a first attempt towards an algorithmic decomposition theorem for quasi-line graphs: the decomposition was not as efficient as more recent works, but thanks to it the authors could develop the first algorithm for the MWSS on claw-free graphs that does not use augmenting paths techniques. Finally in [17] Faenza, Oriolo and Stauffer proposed the following algorithmic decomposition theorem for quasi-line graphs.

Theorem 4.2. *[17] Let $G(V, E)$ be a connected quasi-line graph. In time $O(|V||E|)$, one can:*

(i) either recognize that G is net-free;

(ii) or provide a decomposition of G into $k \leq |V|$ quasi-line strips $(G^1, \mathcal{A}^1)$, $\ldots, (G^k, \mathcal{A}^k)$, with respect to a partition $\mathcal{P}$, such that each graph G^i is distance simplicial with respect to each clique $A \in \mathcal{A}^i$.

Moreover, if $\mathcal{A}^i = \{A_1^i, A_2^i\}$, then:

- *either $A_1^i = A_2^i = V(G^i)$;*
- *or $A_1^i \cap A_2^i = \emptyset$ and there exits j_2 such that $A_2^i \cap N_{j_2}(A_1^i) \neq \emptyset$, $A_2^i \subseteq N_{j_2-1}(A_1^i) \cup N_{j_2}(A_1^i)$ and $N_{j_2+1}(A_1^i) = \emptyset$, where $N_j(A_1^i)$ is the j-th neighborhood of A_1^i in G^i (and, analogously, there exits j_1 such that $A_1^i \cap N_{j_1}(A_2^i) \neq \emptyset$, $A_1^i \subseteq N_{j_1-1}(A_2^i) \cup N_{j_1}(A_2^i)$ and $N_{j_1+1}(A_2^i) = \emptyset$). Besides, each vertex in A has a neighbor in $V(G^i) \setminus A$, for each clique $A \in \mathcal{A}^i$. Finally, if $A_1^i \cap A_2^i$ are in the same set of $\mathcal{P}$, then A_1^i is anticomplete to A_2^i.*

This theorem is indeed a structure theorem, because it says that either G belongs to a class of graphs (claw and net-free graphs), or it is the composition of some strips, but the most interesting thing is that we can say if G falls in case (i) or (ii) in $O(|V||E|)$ time. Again this theorem was exploited to obtain an algorithm for the MWSS problem, with a relevant improvement of the running time.

We will call a connected claw-free perfect graph G *decomposable* if it falls in case (ii) of Theorem 4.2. Decomposable graphs are strip-composed graphs, moreover Theorem 4.2 states that there is an algorithm that provides the decomposition in strips. In Section 4.4 we describe an algorithm for the MWCC on decomposable graphs.

4.3 Related work: an algorithm for the MWCC problem on claw-free perfect graphs

Before the structure of claw-free graphs was deeply investigated, Hsu and Nemhauser in [23] have presented the first combinatorial algorithm for the MWCC problem on claw-free perfect graphs. It was already known that the problem was polynomially solvable on all perfect graphs using an algorithm by Groetschel Lovász and Schrijver (see [21], this algorithm is not combinatorial), but there was an interest on finding some combinatorial algorithms for subclasses of perfect graphs. The algorithm in [23] uses the property that in perfect graphs we always have a *crucial* clique:

Definition 4.3. *Let $G(V, E)$ be a graph with a weight function on the vertices $w : V \to \mathbb{R}^+$. A clique K of G is crucial if and only if $K \cap S \neq \emptyset$ for every* MWSS *S of G.*

Observe that once we have a crucial clique C we can find in polynomial time a suitable δ such that $y_C = \delta$, $w'(v) = w(v) - \delta$ for every $v \in C$ and $w'(v) = w(v)$ for every $v \in V \setminus C$ and $\alpha_{w'}(G) = \alpha_w(G) - \delta$ (we can find in polinomial time such a δ because the graph is perfect, thus we can compute a MWSS in polynomial time). Hence if we have a combinatorial algorithm that finds a crucial clique in a perfect graph G, we can iteratively find such a clique and a suitable δ, till every vertex has weight zero (or if we delete vertices with weight zero, till the graph G is empty). This gives a simple pseudopolynomial algorithm for all perfect graphs.

Hsu and Nemhauser do in their algorithm for claw-free perfect graphs something very similar to the pseudopolynomial procedure that we have just sketched for the whole class of perfect graphs. They find a MWSS S of G

and then they select an $s \in S$ and find a family of crucial cliques $\mathcal{K}_s$ containing s such that $\sum_{K \in \mathcal{K}_s} y_K = w(s)$ and such that the weighted stability number $\alpha_w(G)$ has dropped down by quantity $\sum_{K \in \mathcal{K}_s} y_K$. With these two conditions they know that $S \setminus \{s\}$ is a maximum weighted stable set in G where the vertices have a new weight function $w'(v) = w(v) - \sum_{K \in \mathcal{K}_s : v \in K} y_K$, and they can iterate the procedure.

The key fact is that in order to find a crucial clique in $N(s)$ they need only to understand which vertices or subset of vertices in $N(s)$ belong to a MWSS of G, and to do this they use Minty's augmenting path algorithm for the MWSS [31] (which has a flaw that has been corrected by Nakamura and Tamura, see [32], or can be avoided via preprocessing, as Schrijver shows in [41]). Moreover, thanks to the quasi-liness of the graph, they can show that the algorithm is polynomial.

We can conclude that the algorithm of Hsu and Nemhauser is essentially a "dual" algorithm as it relies on any algorithm for the MWSS problem in claw-free graphs (we have, nowadays, several algorithms for this, see [31, 32, 34, 17, 33, 41]), and, in fact, builds a MWCC by a clever use of linear programming complementarity slackness. The computational complexity of the algorithm by Hsu and Nemhauser is $O(|V(G)|^{5.5})$. To the best of our knowledge, this is so far the only available combinatorial algorithm to solve the MWCC in claw-free perfect graphs.

We observe that the algorithm proposed by Hsu and Nemhauser needs to compute at the beginning a MWSS of the whole graph, and then it needs to compute many augmenting paths. We would like to have an algorithm which either avoids at all the computation of MWSS of the graph G, or it builds at the same time a MWCC and a MWSS of G, using for this computation a routine for the unweighted versions of the problems.

4.4 An algorithm for the MWCC on decomposable graphs

In this section we present an algorithm for the MWCC problem on decomposable graphs. This algorithm is basically the algorithm for strip-composed perfect graphs of Chapter 3, customized on decomposable graphs, that, thanks to Theorem 4.2, have a special structure of the strips. We will use the same notation of Chapter 3, in particular $\tilde{G}$ will be the composition of the gadgets w.r.t. the partition $\tilde{\mathcal{P}}$ and H is the root graph of $\tilde{G}$.

If we are interested in finding a MWCC of G, following Theorem 3.5, we must show that for a strip that is distance simplicial we can compute in polynomial time: a MWCC of the strip; a MWCC of $G^i_\bullet$, i.e. G^i plus a vertex complete to both extremities, when the strip $(G^i, \mathcal{A}^i)$ is odd-short; a MWCC of $G^i_=$, i.e. G^i plus the edges joining the extremities A^i_1, A^i_2 of the strip, when they are in the same class of the partition and there is an A^i_1–A^i_2 path of length two. Before getting into these details we underline that some of the results in Chapter 3 are not necessary.

This is the case for Lemma 3.24. Suppose that in a decomposable graph we have a multi-triangle x_j, x_ℓ, x_k in H such that: there exist $\tilde{H}^3_a, \tilde{H}^3_b, \tilde{H}^3_c$ with the extremities of $\tilde{H}^3_a$ in P_j and P_ℓ, the extremities of $\tilde{H}^3_b$ in P_ℓ and P_k, the extremities of $\tilde{H}^3_c$ in P_k and P_j. By construction, each of the strip H_a, H_b, H_c is either an even strip or an even-odd strip $(G^i, \mathcal{A}^i)$, for $i \in \{a, b, c\}$. Let a_1, a_2 be the endpoints of an even A^a_1–A^a_2 path, and define b_1, b_2 and c_1, c_2 analogously. Then these three paths along with the edges a_2b_1, b_2c_2 and a_1c_1 induce an odd hole, unless the three paths have length zero, i.e, $a_1 = a_2$, $b_1 = b_2$ and $c_1 = c_2$. That is the case when G is a perfect graph. If $a_1 = a_2$, $b_1 = b_2$ and $c_1 = c_2$ then $A^a_1 \cap A^a_2 \neq \emptyset$, $A^b_1 \cap A^b_2 \neq \emptyset$ and $A^c_1 \cap A^c_2 \neq \emptyset$. Then from Theorem 4.2, as G is a decomposable perfect graph, $A^a_1 = A^a_2 = G^a$, $A^b_1 = A^b_2 = G^b$ and $A^c_1 = A^c_2 = G^c$.

So, to the clique of $\tilde{G}$ corresponding to the (multi)triangle $x_i x_j x_\ell$ in H, we will assign in G the clique induced by $\bigcup_{d \in I_{ij\ell}} G^d$, where $I_{ij\ell}$ is the set of indices d of 2-strips in the decomposition, that have been replaced by $\tilde{H}^3_d$, and having their two extremities belonging to two different sets from $\{P_i, P_j, P_\ell\}$. It follows that we do not need Lemma 3.24, because in a strip H_i with $A^i_1 = A^i_2$, G^i is a clique, $\delta^i_1 = 0$, $\tau_w(G^i \setminus A^i_2) = 0$ and $\tau_w(G^i \setminus A^i_1) = 0$. Thus all the vertices of G^i are already covered by the MWCC of $\tilde{G}$.

Next we want to show that we can solve in polynomial time the MWCC in each strip, or in the other graphs required by Theorem 3.5. In order to show this we have to distinguish two cases: (a) 1-strips and 2-strips with extremities in different classes of the partition $\mathcal{P}$ without a vertex complete to both extremities, (b) 2-strips with extremities in the same class of the partition and a path of length two between the extremities and odd-short 2-strips with extremities in different classes of the partition $\mathcal{P}$ and a vertex complete to both extremities.

For case (a) , from Theorem 4.2 we know that the G^i corresponding to the strip H_i is distance simplicial w.r.t. both extremities (or w.r.t. the unique extremity for 1-strips). We will describe how to compute a MWCC in graphs distance simplicial w.r.t. a clique in next section.

For case (b) we know again that the graph G^i corresponding to the strip H_i is distance simplicial w.r.t. the extremities (and thus we can find a MWCC of such graphs as we do for graphs falling in case (a)), but we cannot say that the graphs we obtain adding to the strip the edges between the extremities or a vertex complete to both extremities fall in this class. We can nevertheless find in polynomial time a MWCC of such graphs. We give more details in Section 4.6.

4.5 Computing a MWCC on a graph distance simplicial w.r.t. a clique K

From Theorem 4.2 we know that the graphs corresponding to strips that fall in case (a) are distance simplicial w.r.t. both extremities (or w.r.t. the unique extremity for 1-strips). For graphs distance simplicial w.r.t. some clique K we have designed an algorithm that does not need the computation of any MWSS of the graph. We start with some definitions and some easy propositions, then we outline an algorithm for finding a MWCC in a distance simplicial graph $G(V, E)$ with a weight function $w : V \to \mathbb{R}^+$ on the vertices.

Definition 4.4. *We say that a clique K of a connected graph G is* distance simplicial *if, for every j, $\alpha(N_j(K)) \leq 1$. In this case, we also say that G is distance simplicial with respect to K.*

We assume therefore that G has a clique K such that for every $1 \leq j \leq t$,

$\alpha(N_j(K)) = 1$ and $N_{t+1}(K) = \emptyset$. We let K_1 be this clique and let $K_{j+1} := N_j(K_1)$, for every $1 \leq j \leq t$.

The following propositions are trivial (note that Propositions 4.5, 4.6 and 4.7 hold as soon as a graph has some clique K such that $N(K)$ is a clique: for our distance simplicial graph K_1 is such a clique).

Proposition 4.5. *Let S be a* MWSS *S of G. Then $S \cap (K_1 \cup N(K_1)) \neq \emptyset$.*

Proposition 4.6. *Let S be a* MWSS *S of G. If $S \cap K_1 = \emptyset$, then $\exists\ s \in S \cap N(K_1)$ complete to K.*

Proposition 4.7. *Let $K := K_1 \cup \{v \notin K_1 : v$ is complete to $K_1\}$. Then K is a clique that intersects every* MWSS *of G.*

In particular it follows from proposition 4.7 that K is a crucial clique (see definition 4.3).

Algorithm 3

Require: A graph $D(V, E)$ that is distance simplicial graph w.r.t. a clique K_1 and a weight function $w : V \to \mathbb{R}^+$.
(Assume $K_{j+1} := N_j(K_1) \neq \emptyset$, for every $1 \leq j \leq t$, $N_{t+1}(K_1) = \emptyset$).

Ensure: A MWCC for $D(V, E)$ w.r.t. the weight function w.

1: Let $i \leftarrow 1$; $Q \leftarrow V$; $y = 0$;
2: While $Q \neq \emptyset$ let $D \leftarrow D[Q]$ and do:
  2.1 Let $j \in [t]$ be such that $K_1 \cap Q = \ldots = K_{j-1} \cap Q = \emptyset$ and $K_j \cap Q \neq \emptyset$.
  2.2 Let $K \leftarrow K_j \cup \{v \notin K_j : v$ is complete to K_j in the graph $D[Q]\}$.
  2.3 Let $\bar{v}$ be the vertex of K with minimum (current) weight w.
  2.4 Let $Q \leftarrow Q \setminus \{v \in K : w(v) = w(\bar{v})\}$.
  2.5 For each $v \in K$, let $w(v) \leftarrow w(v) - w(\bar{v})$.
  2.6 Let $y_K \leftarrow w(\bar{v})$.
3: **Return** y.

Lemma 4.8. *Algorithm 3 is correct and can be implemented as to run in $O(|V(D)|^2)$-time.*

Proof. We claim the following property.

Claim 4.9. *Let $Q \subseteq V(D)$ be a nonempty subset of vertices. Let $j \in \{1,\ldots,t+1\}$ be such that $K_i \cap Q = \emptyset$ for every $1 \leq i < j$, and $K_j \cap Q \neq \emptyset$. Then, in $D[Q]$, $(K_j \cap Q) \cup \{v \in Q \setminus K_j : v$ is complete to K_j in the graph $D[Q]\}$ is a crucial clique.*

Proof. Since D is distance simplicial w.r.t. K_1, $(K_j \cap Q) \cup \{v \in Q \setminus K_j : v$ is complete to $K_j \cap Q$ in the graph $D[Q]\}$ is a clique in $D[Q]$. Suppose that there is a MWSS S in $D[Q]$ that does not intersect it. In particular, $j < t+1$, no vertex of S belongs to K_j, and no vertex of S is complete to $K_j \cap Q$. Since K_{j+1} is a clique, at most one vertex S belongs to it, and any other vertex of S is anticomplete to K_j. In any case, there is a vertex in K_j that is anticomplete to S, a contradiction to the maximality of S, since the weight w is strictly positive. This proves the claim. □

By the claim, the set K we build at step 2.2 is a clique that intersects every MWSS of the current graph. In steps 2.5 and 2.6 we are decreasing the weighted stability number of the current graph by $y(K)$, or in other words $\sum_{K \in \mathcal{K}(D)} y(K) = \alpha_w(D)$ (where $\mathcal{K}(D)$ is the collection of all the cliques of the graph D). In fact let us call w' the weight function after step 2.5 and suppose by contradiction that $\alpha_{w'}(D) > \alpha_w(D) - \bar{w}$, and denote with S' the maximum weight stable set w.r.t. the weight function w' and with $D' := D[Q]$ after step 2.4. We have to analyze two cases: (i) $S' \cap K \neq \emptyset$ and (ii) $S' \cap K = \emptyset$. If (i) holds then $\alpha_{w'}(D) = w'(S') = w(S') - \bar{w} \leq \alpha_w(D) - \bar{w}$ which is a contradiction. If (ii) holds we know that in D', $\bar{K} = \bar{K_j} \cup \{v \notin \bar{K_j} :$ v is complete to $\bar{K_j}$ in the graph $D[Q]\}$, where $\bar{K_j}$ is K_j restricted to vertices with strictly positive weight, is a crucial clique, so in particular $S' \cap \bar{K} \neq \emptyset$. Since $S' \cap K = \emptyset$ and $S' \cap \bar{K} \neq \emptyset$, we have that S' contains a vertex $x \notin \bar{K_j}$ such that x is complete to $\bar{K_j}$ in the graph $D[Q]$ that in D was not complete to K_j, or in other words, x was not adjacent to some vertex $z \in K_j$ of weight $\bar{w}$. But then we can consider the set $S' \cup \{z\}$ and we can observe that this is a stable set in D, and its weight is $w'(S') + \bar{w} = \alpha_{w'}(D) + \bar{w} > \alpha_w(D)$, which is a contradiction.

Moreover, as the stop condition for step 2 is $Q = \emptyset$, we have covered every vertex with its weight and this concludes correctness.

It is trivial to observe that steps 2.1 to 2.6 can be implemented as to run in $O(|V(D)|)$-time, and we can easily observe that they will be repeated at most $|V(D)|$ because each time we perform step 2.4 the cardinality of the set Q strictly decreases. □

Thanks to Algorithm 3 we can compute a MWCC of G^i for every $i = 1, \ldots, k$. We are left to show that we can compute also a MWCC of the graph induced by an odd-short 2-strip with a vertex complete to both extremities and that we can compute a MWCC of the graph induced by a 2-strip with the two extremities in the same class of $\mathcal{P}$ and the edges connecting the two extremities, when there is a path of length two between those two extremities.

4.6 Computing a MWCC for strips in case (b)

In this section we show that we can compute in polynomial time a MWCC for the graphs $G^i_\bullet$ and $G^i_=$ when required from Theorem 3.5.

We start with graphs obtained from odd-short 2-strips H_i plus a vertex complete to both extremities, that we denote again with $G^i_\bullet$. We have seen in Section 3.7 of Chapter 3 that we may need to cover these graphs when we have a multitriangle $y^i_{jl}x_jx_\ell$ in the root graph of $\tilde{G}$, in order to reconstruct a MWCC of G from a MWCC of $\tilde{G}$. We have also seen that in this case the strip H_i is odd-short and by claim 3.20, $G^i_\bullet$ is an induced subgraph of G. It follows that $G^i_\bullet$ is claw-free and perfect: in next lemma we show that under this conditions (claw-freeness and perfection of $G^i_\bullet$) the graph we obtain adding a vertex complete to both extremities of and odd-short strip is cobipartite.

Lemma 4.10. *Let $H_i = (G^i, \mathcal{A}^i)$ be an odd-short strip satisfying condition (ii) of Theorem 4.2, and such that the graph $G^i_\bullet$ obtained from G^i by adding a new vertex x complete to both A^i_1 and A^i_2 is claw-free and perfect. Then, $G^i_\bullet$ is the complement of a bipartite graph.*

Proof. Since $G^i_\bullet$ is odd-short then $A^i_2 \cap N(A^i_1) \neq \emptyset$. As H_i satisfies condition (ii) of Theorem 4.2, $N_3(A^i_1)$ is empty, either $N_2(A^i_1)$ is empty or $A^i_2 \cap N_2(A^i_1) \neq \emptyset$, and $A^i_2 \subseteq N(A^i_1) \cup N_2(A^i_1)$. If $N_2(A^i_1)$ is empty, then $(A^i_1 \cup \{x\}, N(A^i_1))$ is a bipartition of $\overline{G^i_\bullet}$. The same holds if $N_2(A^i_2)$ is empty. So, suppose that $N_2(A^i_1)$ and $N_2(A^i_2)$ are both nonempty.

We claim that

(i) $N(A^i_1) \setminus A^i_2 = N(A^i_2) \setminus A^i_1$

(ii) $N_2(A^i_1) \setminus (A^i_1 \cup A^i_2) = \emptyset$

Proof of Claim (i). Let $v \in N(A^i_1) \setminus A^i_2$ then, since G^i is distance simplicial with respect to A^i_1, v is complete to $A^i_2 \cap N(A^i_1)$, that is nonempty. And so, $v \in N(A^i_2)$. Symmetrically, every vertex in $N(A^i_2) \setminus A^i_1$ belongs to $N(A^i_1)$, and that proves the claim.

Proof of Claim (ii). Suppose there is a vertex $v \in N_2(A^i_1) \setminus A^i_2$. Then, since G^i is distance simplicial with respect to A^i_1, v is complete to $A^i_2 \cap N_2(A^i_1)$, that is nonempty. And so, $v \in N(A^i_2)$. But, by claim (i) v would then belong to $N(A^i_1)$, a contradiction.

In particular, Claims (i) and (ii) imply that $B = V(G^i) \setminus (A^i_1 \cup A^i_2) \subseteq N(A^i_1)$ is a clique.

The vertices of $G^i_\bullet$ can be partitioned into four cliques, namely A^i_1, A^i_2, $\{x\}$, and B, such that $\{x\}$ is complete to $A^i_1 \cup A^i_2$, and B is complete to $(N(A^i_1) \cap A^i_2) \cup (N(A^i_2) \cap A^i_1)$. Moreover, by Theorem 4.2, each vertex in $(N_2(A^i_1) \cap A^i_2) \cup (N_2(A^i_2) \cap A^i_1)$ has a neighbor in B. In particular, since $N_2(A^i_1) \cap A^i_2$ is nonempty, B is nonempty.

Since $G^i_\bullet$ is perfect, in order to prove that it is the complement of a bipartite graph, it is enough to prove that it has no stable set of size three. Since the non-neighbors of x form a clique, if there is a stable set of size 3, then it has one vertex in each of $A^i_1 \cap N_2(A^i_2)$, $A^i_2 \cap N_2(A^i_1)$, and B. Let v, v' be two nonadjacent vertices in A^i_1 and A^i_2, respectively. Then, they cannot have both a common neighbor and a common non-neighbor in B. To the contrary, let w be a common neighbor and w' a common non-neighbor of v, v' in B. Since B is a clique, w, w', v, v' induce a claw in G^i, a contradiction. Suppose that v, v' have a common non-neighbor in B. Since they have also at least one neighbor each in B, and they do not have a common neighbor, there exist $w, w' \in B$ such that w is adjacent to v and not to v' and w' is adjacent to v' and not to v. But then $vww'v'x$ induce a hole of length five on $G^i_\bullet$, a contradiction. So, there is no stable set of size three in $G^i_\bullet$, and it is the complement of a bipartite graph. □

Let G be a cobipartite graph and let K_1 and K_2 be two cliques of G, with $K_1 \cup K_2 = V$. Then G is distance simplicial w.r.t. K_1 and w.r.t. K_2 and we can use algorithm 3 to find a MWCC of G.

Finally, we need to face the case of 2-strips with both extremities in the same class of the partition $\mathcal{P}$. We observe that the graphs induced by these strips are distance simplicial w.r.t. each one of the extremities but the graph we obtain adding to those strips the edges between the extremities in general it is not distance simplicial w.r.t. one of the extremities.

First we recall from the previous chapter that when we have to compute the weight function on the vertices of the gadget associated to 2-strips with both extremities in the same class of the partition $\mathcal{P}$, we never need to consider the edges between the extremities, thus for that purpose we can treat these 2-strips as 2-strips with extremities in different classes of the partition and use the algorithm presented in Section 4.5.

Hence we need only to show that we can compute in polynomial time a MWCC of $G^i_=$ induced by a graph G^i corresponding to a 2-strip H_i plus the edges between the extremities of value $\delta^i_1 = \alpha_w(G^i \setminus (A^i_1 \cup A^i_2))$, which is the value we are left to cover after finding a MWCC of $\tilde{G}$. From Lemma 3.22 we know that we can neglect the edges between the extremities only when the two extremities do not have any common neighbour in $V(G^i) \setminus (A^i_1 \cup A^i_2)$. In next lemmas we are going to show how to deal with the case when there are two vertices $v_1 \in A^i_1$ and $v_2 \in A^i_2$ having a common neighbor in $V(G^i) \setminus (A^i_1 \cup A^i_2)$.

Lemma 4.11. *Let $H_i = (G^i, \mathcal{A}^i)$ be a 2-strip satisfying condition (ii) of Theorem 4.2, and such that there are two vertices $v_1 \in A^i_1$ and $v_2 \in A^i_2$ having a common neighbor in $V(G^i) \setminus (A^i_1 \cup A^i_2)$. Let w be a strictly positive weight function defined on the vertices of G^i, and let $\delta^i_1 = \alpha_w(G^i \setminus (A^i_1 \cup A^i_2))$. Let $G^i_=$ be the graph obtained from G^i by adding the edges between*

A_1^i and A_2^i. Suppose that $\alpha_w(G^i_=) = \delta_1^i$ and that $G^i_=$ is perfect and claw-free. Then, $V(G^i_=) \setminus (A_1^i \cup A_2^i)$ can be partitioned into three complete sets, namely $B = (N(A_1^i) \setminus A_2^i) \cap (N(A_2^i) \setminus A_1^i)$, $C_1 = N(A_1^i) \setminus (A_2^i \cup N(A_2^i))$ and $C_2 = N(A_2^i) \setminus (A_1^i \cup N(A_1^i))$. Moreover, B is complete to $C_1 \cup C_2$, A_1^i is anticomplete to C_2 and A_2^i is anticomplete to C_1.

Proof. Let us consider now the graph G^i that, by Theorem 4.2, is distance simplicial with respect to A_1^i and A_2^i and in which, by the same theorem, A_1^i is anticomplete to A_2^i. By hypothesis, there are two vertices $v_1 \in A_1^i$ and $v_2 \in A_2^i$ having a common neighbor in $V(G^i) \setminus (A_1^i \cup A_2^i)$. So, $B = N^{G^i}(A_1^i) \cap N^{G^i}(A_2^i) = (N^{G^i_=}(A_1^i) \setminus A_2^i) \cap (N^{G^i_=}(A_2^i) \setminus A_1^i)$ is non-empty. So, there is a vertex in $A_2^i \cap N_2^{G^i}(A_1^i)$ and, by Theorem 4.2, ${N_4}^{G^i}(A_1^i)$ is empty. Symmetrically, ${N_4}^{G^i}(A_2^i)$ is empty. Let $C_1 = N^{G^i}(A_1^i) \setminus N^{G^i}(A_2^i) = N^{G^i_=}(A_1^i) \setminus (A_2^i \cup N^{G^i_=}(A_2^i))$ and $C_2 = N^{G^i}(A_2^i) \setminus N^{G^i}(A_1^i) = N^{G^i_=}(A_2^i) \setminus (A_1^i \cup N^{G^i_=}(A_1^i))$. Since G^i is distance simplicial with respect to A_1^i and A_2^i, B is a clique and it is complete to C_1 and C_2. Moreover, $N^{G^i}(A_1^i)C_1 \cup B$, and $N^{G^i}(A_2^i) = C_2 \cup B$. Since B is non-empty, $A_2^i \cap N_2^{G^i}(A_1^i)$ is non-empty. Since $N_2^{G^i}(A_1^i)$ is a clique, $N_2^{G^i}(A_1^i) \subseteq (A_2^i \cup N^{G^i}(A_2^i)) \setminus N^{G^i}(A_1^i) = A_2^i \cup C_2$. Symmetrically, $N_2^{G^i}(A_2^i) \subseteq (A_1^i \cup N^{G^i}(A_1^i)) \setminus N^{G^i}(A_2^i) = A_1^i \cup C_1$. Suppose that $N_3^{G^i}(A_1^i)$ is non-empty, and let $v \in N_3^{G^i}(A_1^i)$. Then v has a neighbor in $N_2^{G^i}(A_1^i) \subseteq A_2^i \cup N^{G^i}(A_2^i)$, thus $v \in A_2^i \cup N^{G^i}(A_2^i) \cup N_2^{G^i}(A_2^i) \subseteq A_2^i \cup C_2 \cup B \cup A_1^i \cup C_1 \subseteq A_1^i \cup N^{G^i}(A_1^i) \cup N_2^{G^i}(A_1^i)$, a contradiction. Therefore, $N_3^{G^i}(A_1^i)$ and $N_3^{G^i}(A_2^i)$ are empty, and the lemma holds. ☐

Lemma 4.12. *Let $H_i = (G^i, \mathcal{A}^i)$ be a 2-strip satisfying the conditions of Lemma 4.11, and let B, C_1, C_2, $G^i_=$ be defined as there. Let w be a strictly positive weight function defined on the vertices of G^i, and let $\delta_1^i = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$. Suppose that $\alpha_w(G^i_=) = \delta_1^i$ and that $G^i_=$ is perfect and claw-free. Then, either $G^i_=$ is the complement of a bipartite graph, or there exists a* MWCC *of G^i that is also a* MWCC *of $G^i_=$. In particular, $\alpha_w(G^i) = \delta_1^i$.*

Proof. Suppose that there is no MWCC of G^i that is also a MWCC of $G^i_=$. Then, every MWCC of $G^i_=$ contains a clique $\mathcal{C}$ that is not a clique of G^i, thus, it intersects both A_1^i and A_2^i and, since A_1^i is anticomplete to C_2 and

A_2^i is anticomplete to C_1, $\mathcal{C} \subseteq A_1^i \cup A_2^i \cup B$. Since $\mathcal{C}$ is a crucial clique of $G_{=}^i$ (it has positive weight in a MWCC of $G_{=}^i$), $\mathcal{C}$ intersects every maximum weight stable set of $G_{=}^i$. In particular, since $\alpha_w(G_{=}^i) = \alpha_w(G^i \setminus (A_1^i \cup A_2^i))$, it intersects every maximum weight stable set of $G^i \setminus (A_1^i \cup A_2^i)$. So, there is a maximum stable set S of $G^i \setminus (A_1^i \cup A_2^i)$ such that $S \subseteq B$, namely, $S = \{b\}$, with $b \in B$. Since $\{b\}$ is also a maximum stable set of $G_{=}^i$ and w is strictly positive, b is complete to $V(G_{=}^i) \setminus \{b\}$. Finally, a quasi-line graph containing a universal vertex is complement bipartite. □

If $G_{=}^i$ is cobipartite, then we can compute the MWCC as we described before. If it is not again we may simply ignore the edges between the two extremities of the strip and then compute a MWCC in G^i, which is distance simplicial w.r.t. each one of the extremities, using Algorithm 3.

Finally we can prove the following theorem for decomposable graphs

Theorem 4.13. *Let $G(V, E)$ be a claw-free perfect graph with a weight function on the vertices $w : V \to \mathbb{R}^+$ and let G be as in case (ii) of Theorem 4.2. Then we can compute a* MWCC *of G w.r.t. w in time $O(|V(G)|^3)$, using Algorithm 2.*

Proof. From Theorem 3.5, we know that, given the decomposition of G in strips, we can compute a MWCC of G in time $O(\sum_{i=1}^k p_i(|V|) + match(|V|))$. For every 2-strip with extremities in different classes of $\mathcal{P}$ and for every 1-strip, from Lemma 4.8 $p_i(|V|) = O(|V(G^i)|^2)$. For every 2-strip with the extremities in the same class of $\mathcal{P}$, we first need to check if $G_{=}^i$ is cobipartite, which takes $O(|V(G^i)| + |E(G_{=}^i)|)$, and then we either compute directly a MWCC of $G_{=}^i$ or we compute a MWCC of G^i, and in both cases it takes $O(|V(G^i)|^2)$. Finally, for the computation of the MWCC of $G_{\bullet}^i$, when needed, it takes again $O(|V(G^i)|^2)$. Then $O(\sum_{i=1}^k p_i(|V|)) = O(|V(G)|^2)$ and the overall complexity Algorithm 2 for the MWCC is $O(|V(G)|^2 + |V(G)|^2 log|V(G)|)$ $= O(|V(G)|^2 log|V(G)|)$ (using the primal dual algorithm for maximum weight matching by Gabow [19])). As it takes $O(|V(G)|^3)$ to obtain the decomposition in strips, this is the overall complexity bound of the algorithm. □

Chapter 5

A fast algorithm to reduce proper and homogeneous pairs of cliques

5.1 Introduction

A pair of vertex-disjoint cliques $\{K_1, K_2\}$ is *homogeneous* if every vertex that is neither in K_1, nor in K_2 is either adjacent to all vertices from K_1, or non-adjacent to all of them, and similarly for K_2. Homogeneous pairs of cliques were first defined in the context of bull-free graphs [11], and seem to play a non-trivial role in combinatorial, structural and polyhedral properties of claw-free graphs. For instance, a well-known decomposition result by Chudnovsky and Seymour is as follows:

Theorem 5.1. *[8] For every connected claw-free graph G with $\alpha(G) \geq 4$, if G does not admit a 1-join and there is no homogeneous pair of cliques in G, then either G is a circular interval graph, or G is a composition of linear interval strips, XX-strips, and antihat strips.*

See [8] for the definition of graphs and operations involved in Theorem 5.1: we skip them, since they are of no use here. What is interesting to us is the fact that homogeneous pairs of cliques are somehow an *annoying* structure: as it is written in [8], "There is also a “fuzzy” version of this (*i.e.*

Theorem 5.1), without the hypothesis that there is no homogeneous pair of cliques in G, but it is quite complicated". (This more complex version of the theorem is actually given in [9].) A similar situation can be found in the structure theorem on Berge graphs [7].

In the literature, some effort has been devoted to design *reduction techniques* to get rid of homogeneous pairs of cliques that are also *proper*. We say that a pair of cliques $\{K_1, K_2\}$ is *proper* if each vertex in K_1 is neither complete nor anticomplete to K_2, and each vertex in K_2 is neither complete nor anticomplete to K_1. Those reduction techniques are designed to preserve graph invariants, such as chromatic number [25, 24] and stability number [34], or graph properties, such as the property of a graph of being quasi-line [5], fuzzy circular interval [35], or even facets of the stable set polytope [15]. The state of the art complexity for recognizing whether a graph $G(V, E)$ has some proper and homogeneous pairs of cliques is $O(|V(G)|^2|E(G)|)$ [24, 38].

In this chapter, we introduce a reduction operation that generalizes and unifies those different techniques. It essentially replaces a proper and homogeneous pair of cliques $\{K_1, K_2\}$ with another pair of cliques $\{A_1, A_2\}$ that is homogeneous but non-proper. A large number of pairs $\{A_1, A_2\}$ can be used in our reduction, and the choice of a particular pair is done depending on some invariant (or property) we want the reduction to preserve. Regardless of this choice and of the number of proper and homogeneous clique of the input graph G, we show that our reduction can be embedded in a fast algorithm that iteratively replaces a proper and homogeneous pair of cliques $\{K_1^i, K_2^i\}$ with a non-proper and homogeneous one $\{A_1^i, A_2^i\}$, and outputs after $|E(G)|$ iterations a graph without proper and homogeneous pairs of cliques. We stress that the algorithm is not graph-class specific, i.e. it works with *any* simple graph in input. Our main result will be then the following:

Theorem 5.2. *Let $G(V, E)$ be a graph. Algorithm 5 builds a sequence of graphs $G = G^0, G^1, \ldots, G^q$, with $q \leq |E(G)|$, such that G^q has no proper and homogeneous pairs of cliques, and each G^i, $i < q$, is obtained from G^{i-1} by replacing a proper and homogeneous pair of cliques $\{K_1^i, K_2^i\}$ with an homogeneous pair of cliques $\{A_1^i, A_2^i\}$. The algorithm can be implemented* *as to run in $O(|V(G)|^2|E(G)| + \sum_{i=1}^{q} p(i))$-time, if, for $i = 1, \ldots, q$, it takes $p(i)$-time to generate $G^{i+1}[A_1^i \cup A_2^i]$, from the knowledge of G^i, K_1^i and K_2^i.*

Combining this theorem with a few results from the literature, we will show some more facts, among which:

- we can reduce in time $O(|V(G)|^{\frac{5}{2}}|E(G)|)$ the coloring problem (resp. the maximum clique problem) on a graph $G(V, E)$ to the same problem on a graph G' without proper and homogeneous pairs of cliques;
- we can reduce in time $O(|V(G)|^2|E(G)|)$ the maximum weighted stable set problem on a graph $G(V, E)$ to the same problem on a graph G' without proper and homogeneous pairs of cliques.

The remainder of the chapter is organized as follows: in Section 5.2 we give some definitions and some preliminary results, in Section 5.3 we define a general algorithm for removing proper and homogeneous pairs of cliques, in Section 5.4 we show how one can tailor the algorithm in order to preserve a desired graph invariant or property.

5.2 Preliminaries

Given a simple graph $G(V, E)$, let $n = |V(G)|$ and $m = |E(G)|$. We recall that we denote by uv an edge of G, while we denote by $\{u, v\}$ a pair of vertices $u, v \in V$. We say that v is *universal* to $u \in V$ if v is adjacent to u and to every vertex in $N(u) \setminus \{v\}$. Let $S \subset V$, then $x \notin S$ is *complete* (resp. *anticomplete*) to S in G if $S \cap N(x) = S$ (resp. $S \cap N(x) = \emptyset$). Finally a C_4 is an induced chordless cycle on four vertices.

Definition 5.3. *Let G be a graph and $\{K_1, K_2\}$ be a pair of non-empty and vertex-disjoint cliques. The pair $\{K_1, K_2\}$ is* homogeneous *if each vertex $z \notin (K_1 \cup K_2)$ is either complete or anti-complete to K_1 and either complete or anti-complete to K_2.*

Definition 5.4. *Let K be a clique of a graph G and let $v \notin K$. v is* proper *to K if v is neither complete nor anti-complete to K, and $P(K)$ is the set of vertices that are proper to K.*

Definition 5.5. *Let G be a graph and $\{K_1, K_2\}$ be a pair of non-empty and vertex-disjoint cliques. The pair $\{K_1, K_2\}$ is* proper *if each vertex $u \in K_1$ (K_2, respectively) is proper to K_2 (K_1). A pair of vertex-disjoint cliques that are proper and homogeneous is also called a* PH pair.

We skip the simple proof of the following lemma.

Lemma 5.6. *Let G be a graph and $\{K_1, K_2\}$ be a homogeneous pair of cliques. Then $\{K_1, K_2\}$ is proper if an only if, for each $i \in \{1, 2\}$ and $x \in K_i$, there exist $y_1, y_2 \in K_i$ (possibly $y_1 = y_2$) such that x is non-universal to y_1 and y_2 is non-universal to x.*

In fact, one can show that for each clique K_i of a proper pair $\{K_1, K_2\}$ there always exist two vertices $x, y \in K_i$ that are non-universal to each other. Namely, we have the following (see Lemma 1 in [15]):

Lemma 5.7. *Let $\{K_1, K_2\}$ be a proper pair of cliques in a graph G. Then $G[K_1 \cup K_2]$ contains C_4 as an induced subgraph.*

Hence, when looking for a PH pair in a graph, one can start from a pair of vertices that are adjacent and not universal to each other, and then determine whether they have a *PH-embedding*, namely:

Definition 5.8. *Let u and v be two adjacent vertices of a graph G. We say that u and v have a* PH-embedding *if they are not universal to each other, and there exists a PH pair of cliques $\{K_1, K_2\}$ such that $u, v \in K_1$. We also denote by PH(G) the set of pairs of vertices of G that have a PH-embedding.*

The next lemma is therefore trivial.

Lemma 5.9. *If no pair of vertices of G have a PH-embedding, then G has no PH pairs of cliques.*

Given two adjacent vertices that are non-universal to each other, a simple algorithm recognizes in $O(n^2)$-time whether they have a PH-embedding. This routine, which we report below, was independently proposed by King and Reed [24] and Pietropaoli [38] (see also [35]). Actually King and Reed designed an algorithm for a slightly different problem: call $\{K_1, K_2\}$ a *non-trivial homogeneous* (NTH) pair of cliques in G if $\{K_1, K_2\}$ is a homogeneous pair of cliques in G, and $G[K_1 \cup K_2]$ has an induced C_4. Lemma 5.7 implies that each PH pair of cliques is a NTH pair of cliques, and one can immediately check that the converse does not always hold. But given a NTH pair of cliques $\{K_1, K_2\}$, one can obtain a PH pair of cliques H_1, H_2 with $H_1 \subseteq K_1$, $H_2 \subseteq K_2$, by iteratively removing from $\{K_1, K_2\}$ vertices that are non-proper to the opposite clique. Thus, in order to find a NTH pair one can look for a PH pair: this is exactly what King and Reed do in [24] (see Section 3).

Algorithm 4 Finding a PH-embedding

Require: A graph G, and a pair of adjacent vertices $\{u, v\}$ that are not universal to each other.

Ensure: A PH-embedding $\{K', K\}$ for $\{u, v\}$, if any.

$K' := \{u, v\}$; $K := P(\{u, v\})$;
while K is a clique and $P(K) \neq K'$ **do**
    $K' := K$, $K := P(K)$;
end while
if K is not a clique **then** there is no PH-embedding for $\{u, v\}$: **stop**.
else $P(K) = K'$ and $\{K, K'\}$ is a PH-embedding for $\{u, v\}$: **stop**.

Theorem 5.10. *[24], [38] It is possible to implement Algorithm 4 as to run in $O(|V(G)|^2)$.*

Besides considering pairs of cliques that are proper and homogeneous, we will also consider pairs of cliques that are homogeneous but non-proper. This leads to the following definition:

Definition 5.11. *Let G be a graph and $\{A_1, A_2\}$ be a pair of non-empty and vertex-disjoint cliques that are not complete to each other. The pair $\{A_1, A_2\}$ is C_4^{free} if $G[A_1 \cup A_2]$ has no induced C_4. A pair of cliques that is C_4^{free} and homogeneous is also called a C_4^{free}*H pair*.*

It follows from Lemma 5.7 that no pair of C_4^{free} cliques is proper. We skip the simple proof of the next lemma.

Lemma 5.12. *Let G be a graph and $\{A_1, A_2\}$ be a pair of non-empty and vertex-disjoint cliques that are not complete to each other. Then $\{A_1, A_2\}$ is*

C_4^{free} *if and only if the following holds: if u and $v \in A_1$ then u is universal to v or v is universal to u (note that this property holds if and only if the same happens with the vertices of A_2).*

The next lemma analyzes the possible intersections between PH and C_4^{free}H pairs of cliques.

Lemma 5.13. *Let $G(V, E)$ be a graph with a PH pair of cliques $\{K_1, K_2\}$ and a C_4^{free}H pair of cliques $\{A_1, A_2\}$. Then $K_1 \cap A_2 = K_2 \cap A_1 = \emptyset$ or $K_1 \cap A_1 = K_2 \cap A_2 = \emptyset$.*

Proof. We start with the following:

Claim 5.14. *$K_i \cap A_1 = \emptyset$ or $K_i \cap A_2 = \emptyset$, for $i = 1, 2$.*

Proof. Without loss of generality, suppose to the contrary that there exist $a \in A_1$ and $b \in A_2$ such that $a, b \in K_1$. Being K_1 proper to K_2, there exist $c, d \in K_2$ (possibly non-distinct) such that $ad, bc \notin E$. We first show that $c, d \notin A_1 \cup A_2$. Note that $d \notin A_1$ and $c \notin A_2$. Now suppose that $d \in A_2$; it follows that $d \neq c$. Since c is adjacent to d and not adjacent to b, and $\{A_1, A_2\}$ is a homogeneous pair, it follows that $c \in A_1$. But then a, b, c, d induce a C_4 on $G[A_1 \cup A_2]$, and therefore neither a is universal to c nor c is universal to a, which is a contradiction to Lemma 5.12. We get an analogous contradiction if we assume that $c \in A_1$.

So $c, d \notin A_1 \cup A_2$; being $ad, bc \notin E$ and $\{A_1, A_2\}$ a homogeneous pair, c is anti-complete to A_2 and d is anti-complete to A_1. Since K_2 is a clique, it follows that $K_2 \cap (A_1 \cup A_2) = \emptyset$. Since $A_1 \cup A_2$ is not a clique, there exist $a' \in A_1$, $b' \in A_2$ such that $a'b' \notin E$. Note that $da' \notin E$ and that $a' \notin K_2$. We now show that $a' \notin K_1$. For, suppose the contrary; then $b' \neq b$ and $b' \notin K_1$, and so b' is proper to K_1 and therefore belongs to K_2, which is a contradiction, since we already argued that $K_2 \cap (A_1 \cup A_2) = \emptyset$.

Hence $a' \notin K_1 \cup K_2$. Since $\{K_1, K_2\}$ is a proper pair, there exists a vertex $e \in K_2$ such that $ea \in E$. Since $K_2 \cap (A_1 \cup A_2) = \emptyset$ and $\{A_1, A_2\}$ is a homogeneous pair, it follows that $ea' \in E$. On the other hand, we observed that $da' \notin E$. But then a' is proper to K_2, contradicting $a' \notin K_1$. (*End of the claim.)*

From the claim, we may assume without loss of generality that $K_1 \cap A_1 = \emptyset$. In this case, the statement follows if $K_2 \cap A_2 = \emptyset$, so suppose that there exists $v_2 \in K_2 \cap A_2$. It again follows from the previous claim that $K_2 \cap A_1 \neq \emptyset$; hence the statement follows if $K_1 \cap A_2 = \emptyset$. So suppose that there exists $v_1 \in K_1 \cap A_2$; since $\{K_1, K_2\}$ is a proper pair, it follows that $v_1, v_2 \in A_2$ are not universal to each other, a contradiction to Lemma 5.12. □

5.3 An algorithm for removing proper and homogeneous pairs

We now define an operation of *reduction* that is crucial. This operation essentially replaces a PH pair of cliques with a C_4^{free}H pair of cliques. The latter pair will be defined through a suitable graph that we call, for shortness, a *non-proper 2-clique*.

Definition 5.15. *A* non-proper 2-clique $H_{\{A_1,A_2\}}$ *is a graph with a C_4^{free} pair of cliques $\{A_1, A_2\}$, such that $V(H_{\{A_1,A_2\}}) = A_1 \cup A_2$.*

Definition 5.16. *Let G be a graph with a PH pair of cliques $\{K_1, K_2\}$. Also let $H_{\{A_1,A_2\}}$ be a non-proper 2-clique graph vertex-disjoint from G. The* PH reduction *of G with respect to $(K_1, K_2, H_{\{A_1,A_2\}})$ returns a new graph $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ defined as follows:*

- *$V(G|_{K_1,K_2,H_{\{A_1,A_2\}}}) = (V(G) \setminus (K_1 \cup K_2)) \cup (A_1 \cup A_2)$;*
- *Let x, y be vertices of $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$. The edge $xy \in E(G|_{K_1,K_2,H_{\{A_1,A_2\}}})$ if and only if one of the following holds:*
  - *$xy \in E(G)$ with $x, y \notin K_1 \cup K_2$;*
  - *$xy \in E(H_{\{A_1,A_2\}})$ with $x, y \in A_1 \cup A_2$;*
  - *$y \in A_1$, $x \notin K_1 \cup K_2$ and x is complete to K_1;*
  - *$y \in A_2$, $x \notin K_1 \cup K_2$ and x is complete to K_2.*

We skip the trivial proof of the following lemma.

Lemma 5.17. *The graph $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ is such that the following properties hold:*

- *$\{A_1, A_2\}$ is a $C_4^{free}H$ pair of cliques;*
- *if $x, y \in A_1$ (resp. $x, y \in A_2$), then x is universal to y or y is universal to x;*
- *if $|K_1| \geq |A_1|$ and $|K_2| \geq |A_2|$, then the graph $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ can be built in time $O(|V(G)|^2)$ and $|V(G|_{K_1,K_2,H_{\{A_1,A_2\}}})| \leq |V(G)|$.*

The following crucial lemma shows that all the PH pairs of $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ are "inherited" by the input graph G.

Lemma 5.18. *Let $\{w_1, w_2\}$ be a pair of adjacent vertices of $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ with a PH-embedding. Then:*

1. *w_1 and w_2 do not both belong to $A_1 \cup A_2$;*
2. *if $w_1, w_2 \notin A_1 \cup A_2$, then $\{w_1, w_2\}$ also admits a PH-embedding in G;*
3. *if $w_1 \in A_1$ (resp. $w_1 \in A_2$) and $w_2 \notin A_1 \cup A_2$, then, for each $a \in K_1$ (resp. $a \in K_2$), $\{a, w_2\}$ admits a PH-embedding in G.*

Proof. Throughout the proof, when referring to vertices of $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$, we call *artificial* the vertices of $A_1 \cup A_2$, and *non-artificial* the others. Moreover, we let $G' = G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ and let $\{K_1', K_2'\}$ be a PH-embedding for $\{w_1, w_2\}$ in G'.

It follows from Lemma 5.17 that $\{A_1, A_2\}$ is a C_4^{free}H pair of cliques of G'. Therefore it follows from Lemma 5.13 that $K_1' \cap A_2 = K_2' \cap A_1 = \emptyset$ or $K_1' \cap A_1 = K_2' \cap A_2 = \emptyset$. Now suppose that $w_1, w_2 \in A_1 \cup A_2$, and recall that, by definition, $w_1, w_2 \in K_1'$. It follows that either $w_1, w_2 \in A_1$, or $w_1, w_2 \in A_2$. Thus, there exist two vertices of A_1 (resp. A_2) that are non-universal to each other, contradicting Lemma 5.17. Therefore w_1 and w_2 do not both belong to $A_1 \cup A_2$, i.e. statement 1 holds.

W.l.o.g. in the following we assume that $K_1' \cap A_2 = K_2' \cap A_1 = \emptyset$. Now define the sets H_1, H_2 of vertices in G as follows: for $i = 1, 2$, if K_i' has no artificial vertices, define $H_i = K_i'$; otherwise $H_i = (K_i' \cap V(G)) \cup K_i$. Note that this implies that $H_1 \cap K_2 = H_2 \cap K_1 = \emptyset$ and that H_1 and H_2 are cliques.

Claim 5.19. *Let $u, v \in K_1'$ (respectively K_2') be two non-artificial vertices of G' such that u is non-universal to v in G'. Then $u, v \in H_1$ (respectively H_2) and u is non-universal to v in G.*

Proof. We prove the statement for $u, v \in K_1'$. Since u, v are non-artificial, $u, v \in H_1$ by definition. By hypothesis, there exists $z \in K_2'$ s.t. $uz \notin E(G')$, $vz \in E(G')$. If z is non-artificial, $z \in H_2$ by definition, thus u is non-universal to v in G. Suppose now z is artificial, then $z \in A_2$, since $K_2' \cap A_1 = \emptyset$. Then by construction v is complete and u anticomplete to K_2 in G, thus u is non-universal to v in G. (*End of the claim.*)

Claim 5.20. *Let $u, v \in K_1'$ (respectively K_2'), and suppose u is artificial and v is not. Then $\{v\} \cup K_1 \subseteq H_1$ (resp. $\{v\} \cup K_2 \subseteq H_2$). Furthermore:*

1. *If u is non-universal to v, then a is non-universal to v for each $a \in K_1$ (respectively K_2).*

2. *If v is non-universal to u, then v is non-universal to a, for each $a \in K_1$ (resp. K_2).*

Proof. We prove the statement for $u, v \in K_1'$. We are assuming that $K_1' \cap A_2 = \emptyset$, hence $u \in A_1$. So by definition, $\{v\} \cup K_1 \subseteq H_1$. Suppose u is non-universal to v: there exists $z \in K_2'$ s.t. $uz \notin E(G')$, $vz \in E(G')$. If z is an artificial vertex, then $z \in A_2$, which implies that v is complete to K_2, while each vertex $a \in K_1$ is proper to K_2. If z is non-artificial, then by construction z is anticomplete to K_1 while $vz \in E(G)$. This shows 1. Now suppose that v is non-universal to u, i.e. there exists $z \in K_2'$ such that $uz \in E(G'), vz \notin E(G')$. If z is an artificial vertex, then $K_2 \subseteq H_2$ and v is anticomplete to K_2; since each vertex $a \in K_1$ is proper to K_2, v is non-universal to a. If z is non-artificial, then z is complete to K_1 in G, while $zv \notin E(G)$; thus, v is non-universal to $a \in K_1$.(*End of the claim.*)

Claim 5.21. $\{H_1, H_2\}$ *is a PH pair of cliques in G.*

Proof. We already observed that H_1 and H_2 are cliques, and it is straightforward to see that $\{H_1, H_2\}$ is a homogeneous pair. So we conclude the proof by showing that H_1 is proper to H_2 (the other case following by symmetry).

We need to show that each vertex $x \in H_1$ has at least one neighbor and at least one non-neighbor in H_2. Recall that $x \notin K_2$. Suppose first that $x \in K_1$; then by construction $K_1 \subseteq H_1$ and K_1' has at least one artificial vertex, say a. Since $\{K_1', K_2'\}$ is a proper pair, it follows from Lemma 5.6 that there exist a vertex $t_1 \in K_1'$ to which a is non-universal, and a vertex $t_2 \in K_1'$ which is non-universal to a. If t_1 or t_2 is artificial, then K_2' intersects A_2 (recall that $a, t_1, t_2 \in A_1$ have the same neighborhood outside K_2') and consequently, by construction, $K_2 \subseteq H_2$; then the statement follows since $\{K_1, K_2\}$ is a proper pair of cliques. Conversely, if both t_1 and t_2 are non-artificial, then, using Claim 5.20, we conclude that in G x is non-universal to t_1 and that t_2 is non-universal to x, and therefore x has at least one neighbor and at least one non-neighbor in H_2.

Suppose now $x \notin K_1$: then, x is a non-artificial vertex of K_1', and since $\{K_1', K_2'\}$ is proper, it follows again from Lemma 5.6 that there exist a vertex $t_1 \in K_1'$ to which x is non-universal, and a vertex $t_2 \in K_1'$ which is non-universal to x. If both t_1 and t_2 are non-artificial, then also in G we have that x is non-universal to t_1 and t_2 is non-universal to x. If t_1 or t_2 is artificial, then thanks to Claim 5.20, we may suitably replace t_1 or t_2 with vertices from K_1 as to get the same conclusion. (*End of the claim.*)

We conclude the proof of the lemma: part 2 holds by Claims 5.19 and 5.21, while part 3 holds by Claims 5.20 and 5.21. $\square$

As we show in the following, if we iterate the reduction of Definition 5.16, we end up, in at most $|E(G)|$ steps, with a graph without PH pairs of cliques. We first need a definition and a simple lemma, going along the same lines of Definition 5.16 and Lemma 5.18. For a graph G, we denote by $\binom{V(G)}{2}$ the set of unordered pairs of vertices of $V(G)$.

Definition 5.22. *Let G and $G' := G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ be as in Definition 5.16, and let $S \subseteq \binom{V(G)}{2}$. The set $S|_{K_1,K_2,H_{\{A_1,A_2\}}} \subseteq \binom{V(G')}{2}$ is the set of pairs*

$\{x, y\}$ *such that one of the following hold:*

- $\{x, y\} \in S$ *and* $x, y \notin A_1 \cup A_2$;
- $x \in A_1$, $y \notin A_1 \cup A_2$ *and* $\{\{a, y\} \mid a \in K_1\} \subseteq S$;
- $y \in A_2$, $x \notin A_1 \cup A_2$ *such that* $\{\{x, a\} \mid a \in K_2\} \subseteq S$.

Corollary 5.23. *Let* G, $G' := G|_{K_1,K_2,H_{\{A_1,A_2\}}}$, S *and* $S' := S|_{K_1,K_2,H_{\{A_1,A_2\}}}$ *be as in Definition 5.16 and Definition 5.22.*

(i) *If* S *is a superset of* $PH(G)$, *then* S' *is a superset of* $PH(G')$.

(ii) *If* $|K_1| \geq |A_1|$ *and* $|K_2| \geq |A_2|$, *then* $|S'| < |S|$ *and* S' *can be built from* S *in time* $O(|V(G)|^2)$.

Proof. (i) Pick any pair $\{w_1, w_2\}$ of vertices of G' which admit a PH-embedding in G': by part (1) of Lemma 5.18, they cannot both belong to $A_1 \cup A_2$. Suppose that $w_1, w_2 \notin A_1 \cup A_2$. Then, by part (2) of Lemma 5.18, $\{w_1, w_2\}$ also have a PH-embedding in G and thus $\{w_1, w_2\} \in S$. Then, by construction, $\{w_1, w_2\} \in S'$. Now, suppose that exactly one of them belongs to $A_1 \cup A_2$, w.l.o.g. w_1, and let first $w_1 \in A_1$; then by part (3) of Lemma 5.18, for each $a \in K_1$, $\{a, w_2\}$ is a pair of vertices with a PH-embedding in G, i.e. $\{\{a, w_2\}, a \in K_1\} \subseteq PH(G) \subseteq S$. Then, by construction, $\{w_1, w_2\} \in S'$. A similar argument works for $w_1 \in A_2$. (ii) The statements holds easily by construction. $\square$

We are now ready to give our algorithm, see Algorithm 5 in the following. Note that it is fully determined, but for the choice of the non-proper 2-clique graph $H_{\{A_1^i, A_2^i\}}$ to be used in each iteration i. In fact, the definition of $H_{\{A_1^i, A_2^i\}}$ will in general depend on G^i, K_1^i and K_2^i: this will be discussed in the next section. Given our previous arguments, it is easy to conclude that Theorem 5.2 correctly predicts the output and the time complexity of Algorithm 5: we skip details.

Let us remark here that in Algorithm 5 we start with a set $S^0 = E(G)$, since we assumed no prior knowledge is available on the pair of vertices of G that are candidate to have a PH-embedding. For specific graphs we may have a better knowledge of those, and consequently start from a set S^0 smaller in size. This may lead to asymptotically faster implementations of Algorithm 5.

Algorithm 5 Eliminating all proper and homogeneous pairs of cliques

Require: A graph G.

Ensure: A graph G^q, without PH pairs of cliques, that is obtained from G by successive PH reductions.

$i := 0$; $G^0 := G$; $S^0 := E(G)$;
while S^i is non-empty **do**
  pick a pair $\{u, v\} \in S^i$;
  using Algorithm 4 check whether the pair $\{u, v\} \in S^i$ has a PH-embedding in G^i;
  if u, v have a PH-embedding $\{K_1^i, K_2^i\}$ **then**
    let $H_{\{A_1^i, A_2^i\}}$ be a non-proper 2-clique graph vertex-disjoint from $V(G^0) \cup V(G^1) \cup \ldots \cup V(G^i)$ and such that $|K_1^i| \geq |A_1^i|$ and $|K_2^i| \geq |A_2^i|$;
    $G^{i+1} := G^i|_{K_1^i, K_2^i, H_{\{A_1^i, A_2^i\}}}$ (see Definition 5.16);
    $S^{i+1} := S^i|_{K_1^i, K_2^i, H_{\{A_1^i, A_2^i\}}}$ (see Definition 5.22);
    $i := i + 1$;
  else
    remove the pair $\{u, v\}$ from S^i;
  end if
end while
$q := i$.
return G^q.

5.4 Preserving some graph invariant or property

In this section, we show that suitable PH reductions preserve graph invariants, such as chromatic number, stability number, and clique number, or graph properties, such as perfection, or the property of a graph of being fuzzy circular interval. Most of these reductions were in fact proposed in the literature in specific contexts, but they can actually be embedded in the unifying setting of PH reductions.

In some cases [15, 25, 24, 35] the reductions that were used have the following form: take a PH pair of cliques $\{K_1, K_2\}$ and remove some suitable set of edges between vertices of K_1 and vertices of K_2 so that, in particular, in the resulting graph, no C_4 is contained in the subgraph induced by $K_1 \cup K_2$. In another case [34] the reduction has the following form: take a PH pair of cliques $\{K_1, K_2\}$ and add all possible edges between vertices of K_1 and vertices of K_2 but one. It is easy to show that all those types of reductions can be interpreted in terms of our PH reduction, so we skip such details when presenting them. Therefore, they can be embedded into the iterative framework of Algorithm 5, and one may rely on the complexity bound given by Theorem 5.2.

We begin with a reduction introduced by King and Reed [25, 24] for removing edges in a PH pair of cliques while preserving the chromatic number. Recall that $\chi(G)$ denotes the chromatic, $\chi_f(G)$ the fractional, and $\omega(G)$ the clique number of a graph G.

Lemma 5.24. *[25] Let G be a graph and suppose that we are given a PH pair of cliques $\{K_1, K_2\}$ of G. Also, let X be a maximum clique in $G[K_1 \cup K_2]$, and let G' be the graph obtained from G by removing each edge $uv \in E(G)$ such that: $u \in K_1$; $v \in K_2$; $\{u, v\} \not\subseteq X$. Then:*

(i) G' can be built in time $O(|V(G)|^{\frac{5}{2}})$ (from the knowledge of G, K_1 and K_2);

(ii) $\chi(G) = \chi(G')$, $\chi_f(G) = \chi_f(G')$ and each k-coloring of G' can be extended into a k-coloring of G of in time $O(|V(G)|^{\frac{5}{2}})$.

(iii) $\omega(G) = \omega(G')$, and each clique of G' is also a clique of G.

(iv) If G is claw-free (resp. quasi-line; perfect), then G' is claw-free (resp. quasi-line; perfect).

(One should mention that Lemma 5.24 can be extended to the case where $\{K_1, K_2\}$ is a *nonskeletal* and homogeneous pair of cliques [25]. Also, Andrew King [26] pointed us that this lemma is non-trivially implied by some proofs in [5]. In that paper, Chudnovsky and Ovetsky introduce another reduction for PH pairs of cliques, which is quite similar to the one above. This reduction preserves quasi-liness, while not increasing the clique number of G. It is a simple exercise to show that the reduction in [5] can be interpreted in terms of our PH reduction. Finally, we mention that proposition (*iii*) of Lemma 5.24 is not stated in [25], but it is almost straightforward.)

By embedding the reduction above in the iterative framework of Algorithm 5, we can reduce the problem of computing the chromatic (resp. clique) number on a given graph G to the same problem on a graph G' without PH pairs of cliques.

Corollary 5.25. *From a graph G one can obtain in time $O(|V(G)|^{\frac{5}{2}}|E(G)|)$ a graph G' without PH pairs of cliques such that $\chi(G) = \chi(G')$ and $\omega(G) = \omega(G')$. One can also derive an optimal coloring of G from an optimal coloring in G' in time $O(|V(G)|^{\frac{5}{2}}|E(G)|)$, while a maximum clique in G' is also a maximum clique in G.*

As argued by Li and Zang [28], the maximum weighted clique problem in the complement of a bipartite graph can be reduced to maximum flow, and hence solved in time $O(n^3)$. By building on the latter fact (and slightly increasing the complexity), Corollary 5.25 can be extended to the computation of a graph G' without PH cliques that preserves the maximum weighted clique and its value.

Consider now the maximum weighted stable set problem. Oriolo, Pietropaoli, and Stauffer [34] provide a reduction that preserves the value of a maximum weighted stable set. (We refer to [34] for more details and for the precise definition of the reduction, which is actually stated for the more general class of *semi-homogeneous* pairs of cliques.) By embedding their reduction in Algorithm 5, we obtain the following lemma:

Corollary 5.26. *Let $G(V, E)$ be a graph with a weight function $w : V \mapsto \mathbb{R}$ defined on its vertices. In time $O(|V(G)|^2|E(G)|)$ one can build a graph G' without PH pairs of cliques such that a maximum weighted stable set of G' is also a maximum weighted stable set of G.*

Interestingly, if we now move from the maximum weighted stable set problem to the stable set polytope $STAB(G)$ of a graph G, we can also embed a result in [15] in our framework. Eisenbrand et al. show – see the remark following Lemma 5 in [15] – that each facet of the stable set polytope $STAB(G)$ is also a facet of another graph G' (obtained from G by removing edges) that does not contain any PH pair of cliques. As one easily checks (cfr. the proof of Lemma 5 in [15]), also their result can be phrased in the framework of Algorithm 5.

We now move from graph invariants to graph properties. First, Oriolo, Pietropaoli, and Stauffer [35] show that a suitable reduction of PH pairs of cliques preserves the property of a graph of being, or not being, a fuzzy circular interval graph, and they exploit this fact in an algorithm for recognizing fuzzy circular interval graphs. Their reduction can also be embedded in our framework. In fact, Theorem 5.2 is already used in [35] for bounding the complexity of the recognition algorithm. Finally, every PH reduction preserves perfection, and under very general conditions it does not turn a non-perfect graph into a perfect one. We give just a sketch of the proof of the latter fact, since the arguments used are quite standard.

Lemma 5.27. *Let G be a perfect graph with a PH pair of cliques $\{K_1, K_2\}$. Also let $H_{\{A_1,A_2\}}$ be a non-proper 2-clique graph vertex-disjoint from G. Then the graph $G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ is perfect. The converse implication holds true if A_1 is not anticomplete to A_2.*

Proof. Recall that a graph is perfect if and only if it contains neither odd holes, nor odd antiholes [7]. Let $\{Q_1, Q_2\}$ be a homogeneous pair of cliques in a graph G: it is easy to show that each odd hole (resp. each odd antihole) of G takes at most one vertex from Q_1 and at most one vertex from Q_2.

Suppose first that $G' = G|_{K_1,K_2,H_{\{A_1,A_2\}}}$ is not perfect, i.e. there is an induced subgraph H' of G' that is either a odd hole or an odd antihole. By building on the fact that $|V(H') \cap A_1| \leq 1$ and $|V(H') \cap A_2| \leq 1$, one can easily construct an odd hole (resp. an odd antihole) of G from H', thus showing that G is not perfect as well. Let now A_1 be not anticomplete to

A_2 in G'; then, one can analogously show that if G is not perfect, neither is G'. □

We conclude by pointing out that, with the exception of the reduction from Lemma 5.24 (since $X \subseteq K_1$ or $X \subseteq K_2$ may happen), all the reductions from the current section do not turn an imperfect graph into a perfect one.

Conclusions

In this thesis we mainly studied some combinatorial algorithms for the minimum clique cover (unweighted and weighted) in perfect graphs. Here we want to give an outlook on the possible future directions of research in this topic.

In Chapter 2 we have presented a combinatorial algorithm for the MCC problem on claw-free perfect graphs which relies on the solution of a suitable instance of the 2-SAT problem. We observe here that when we solve such an instance of the 2-SAT problem (or we conclude that the instance is not satisfiable and we find an augmenting path), we are actually asking for an *integer* feasible solution of the following linear system of inequalities (here S is the current stable set):

$$
\begin{array}{ll}
x_{vs} \geq 1 & \text{for every free vertex } v \text{ with } s \in N(v) \cap S \\
x_{us} + x_{vs} \leq 1 & \text{for every } s \in S \text{ and } u, v \in N(s), uv \notin E \\
x_{us_1} + x_{us_2} \geq 1 & \text{for every bound vertex } u \text{ with } \{s_1, s_2\} = N(u) \cap S
\end{array}
$$

where the variable x_{vs} for $s \in S$ and $v \in V \setminus S$ represents how much v is covered from s. An interesting consequence of this observation is that we use the 2-SAT algorithm to test integer feasibility of the previous system, and maybe more efficient techniques can be applied. Moreover it is easy to observe that if there are no free vertices the solution where each variable has value $\frac{1}{2}$ is always a fractional feasible solution of the previous system, thus we can give very easily a trivial solution to the MCC problem on claw-free perfect graphs.

For the MWCC problem on perfect graphs in Chapter 3 we have presented an algorithmic theorem for perfect graphs that are composition of strips and we have described an application of this theorem to claw-free graphs in Chapter 4. Nevertheless it would be interesting to look for other applications of Theorem 3.5 on subclasses of strip-composed perfect graphs (that are not subclasses of claw-free graphs) where the MWCC is easy to compute. In this way, thanks to the machinery presented in Chapter 3 and following the same steps we did for the claw-free case in Chapter 4, one could obtain a polynomial algorithm for the MWCC on those graph classes.

Finally Chapter 4 leaves the following open question: can we find a polynomial time combinatorial algorithm for the MWCC which is more efficient than the algorithm of Hsu and Nemhauser and that can handle a *general* (i.e. not only strip-composed) claw-free perfect graph? We claim that the answer is yes if we can efficiently find an integer feasible solution of the following system of linear inequalities (or conclude that there is no solution and thus find a weighted augmenting path):

$$\begin{array}{ll} x_{vs} \geq w(v) & \text{for every free vertex } v \text{ with } s \in N(v) \cap S \\ x_{us} + x_{vs} \leq w(s) & \text{for every } s \in S \text{ and } u, v \in N(s), uv \notin E \\ x_{us_1} + x_{us_2} \geq w(u) & \text{for every bound vertex } u \text{ with } \{s_1, s_2\} = N(u) \cap S \end{array}$$

where again the variable x_{vs} for $s \in S$ and $v \in V \setminus S$ represents how much v is covered from s. In fact, suppose we have a solution of such a system and let us fix a vertex $s \in S$: we define in $G[N[s]]$ a new weight function $w'_s(s) = w(s)$ and $w'_s(v) = x_{vs}$ for every $v \in N(s)$. As $G[N[s]]$ is cobipartite we can find a MWCC w.r.t. to the weight function w'_s in polynomial time (see the algorithm for graphs that are distance simplicial w.r.t to a clique in Section 4.5), let y_s be such a MWCC. Then it is easy to see that $y = \bigcup_{s \in S} y_s$ is a MWCC of G.

We believe that testing (integer) feasibility in the previous system of linear inequalities can be done in polynomial time with a tecnhique that again produces an auxiliary directed graph, as in the unweighted case. There are non trivial details to be developed regarding how to obtain an augmenting path from a non feasible system and how to make a polynomial number of iterations of this sketched algorithm.

Acknowledgments

This thesis is the outcome of a three years research period mainly spend at the University of Rome "Tor Vergata". In these three years I had the opportunity to meet a lot of friendly and inspiring people.

First of all I would like to thank Gianpaolo and Flavia. I have met Gianpaolo in 2005, when he was teaching a course on graph theory and network flows. Before that course, mathematics for me was mostly represented by functions, derivatives, integrals, and all this stuff that people teaching to engineers love. Any other form of mathematics was mostly a game: Gianpaolo showed me that behind such a game there was a charming and rigorous structure, and I think this is something he continues to teach me every day. I met Flavia when I was working on my master thesis in 2008. I hope that in all this time spent working together I have managed to catch at least a small fraction of her energy and enthusiast approach to graph theory. She is a coauthor of all the research work presented in this thesis except for Chapter 5, and she is also the author of the intricate pictures of Chapter 3.

One of the best experiences of this research period has been the Spring School on Graph Theory held in May 2010 in Montreal. There I finally gave a face to the "famous" Maria Chudnovsky and Paul Seymour. I am really thankful to them for all the time and effort they have spent to try to explain in clear and simple words the massive work they have made on the structure of perfect graphs, claw-free graphs and bull-free graphs.

These three years of work would have never been the same without all the room mates of the Ph.D. room, both for scientific discussion and for fun. Thus I would like to thank Yuri (Faenza, which is a coauthor of the work presented in Chapter 5), Laura and her incomprehensions with the laptop, Enrico and his sister on the phone, Marco and his teletubbies jumpers, Gianmaria "non se po fa un discorso serio", Simone, Matteo, Arianna and Irene. Moreover I would like to thank all the other people of the Operations Research group: Andrea, Sara and Veronica.

Finally I would like to thank my mother and my father for giving me the inspiration for research. Since I remember they brought me to conferences, workshops and laboratories all around the world: I had no idea of the work behind this, but it looked good!

Last but not least I would like to thank Michele: you have been my first and best supporter, understanding my frustrations and encouraging my successes. For all the rest you know, this is just the beginning of a very challenging adventure.

Bibliography

[1] B. Aspvall, M.F. Plass, and R.E. Tarjan "A linear-time algorithm for testing the truth of certain quantified boolean formulas", Information Processing Letters 8, 121-123, (1979).

[2] L.W. Beineke "Characterizations of derived graphs", Journal of Combinatorial Theory 9, 129-135, (1970).

[3] C. Berge "Les problemès de coloration en théorie de graphes", Pubblications de l'Institut de Statistique de l'Université de Paris 9, 123-160, (1960).

[4] C. Berge "Some classes of perfect graphs", in Six papers on Graph Theory, Research and Training School, Indian Statistical Institute, Calcutta, 1-21, (1963).

[5] M. Chudnovsky, and A. Ovetsky "Coloring quasi-line graphs", Journal of Graph Theory 54, 41-50, (2007).

[6] M. Chudnovsky, and M. Plumettaz "The Structure of Claw-Free Perfect Graphs", *submitted for pubblication*, (2011).

[7] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas "The Strong Perfect Graph Theorem", Annals of Mathematics 164, 51-229, (2006).

[8] M. Chudnovsky, and P. Seymour "The Structure of Claw-free Graphs", Surveys in Combinatorics 2005, London Mathematical Society Lecture Note Series 327, 153-171, (2005).

[9] M. Chudnovsky, and P. Seymour "Claw free Graphs V. Global structure", Journal of Combinatorial Theory, Series B 98, 1373-1410, (2008).

[10] M. Chudnovsky, and P. Seymour "Claw-free Graphs VII. Quasi-line graphs", to appear in Journal of Combinatorial Theory, Series B.

[11] V. Chvátal, and N. Sbihi "Bull-free Berge graphs are perfect", Graphs and Combinatorics 3, 127-139, (1987).

[12] V. Chvátal, and N. Sbihi "Recognizing claw-free Berge graphs", Journal of Combinatorial Theory, Series B 44, 154-176,(1988).

[13] J. Edmonds "Paths, trees and flowers", Canadian Journal of Mathematics 17, 449-467, (1965).

[14] J. Edmonds "Maximum matching an a polyhedron with 0,1-vertices" Journal of Research National Bureau of Standards Section B 69, 67-72, (1965).

[15] F. Eisenbrand, G. Oriolo, G. Stauffer and P. Ventura "Circular One Matrices and the Stable Set Polytope of Quasi-Line Graphs", Combinatorica 28(1), 45-67, (2008).

[16] Y. Faenza, G. Oriolo and C. Snels "A fast algorithm to remove proper and homogeneous pairs of cliques (while preserving some graph invariants)", Operations Research Letters 39(3), 213-217, (2011).

[17] Y. Faenza, G. Oriolo and G. Stauffer "An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$-algorithm for the weighted stable set problem", Proceedings of the XXII Annual ACM-SIAM Symposium On Discrete Algorithms, (2010).

[18] D.R. Fulkerson "On the perfect graph theorem", Mathematical Programming, 69-76, (1973).

[19] H.N. Gabow "Data structures for weighted matching and nearest common ancestor with linking", Proceeding of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 321-325, (1990).

[20] M.R. Garey, D.S. Johnson "Computers and Intractability: A Guide to the Theory of NP-completeness", W.H. Freeman, (1979).

[21] M. Groetschel, L. Lovász and A. Schrijver "Geometric Algorithms and Combinatorial Optimization", Springer, Berlin, (1988).

[22] W.L. Hsu, and G.L. Nemhauser "Algorithms for minimum covering by cliques and maximum clique in claw-free perfect graphs", Discrete Mathematics 37 (2-3), 181-191, (1980)

[23] W.L. Hsu, and G.L. Nemhauser "Algorithms for maximum weight cliques, minimum weighted clique covers and cardinality colorings of claw-free perfect graphs", Annals of Discrete Mathematics 21, 317-329, (1984)

[24] A.D. King, and B.A. Reed "Bounding χ in Terms of ω and Δ for Quasi-Line Graphs", Journal of Graph Theory 59, 215-228, (2008).

[25] A.D. King "Claw-free graphs and two conjectures on ω, Δ and χ" PhD Thesis, Mc Gill University, Montreal, (2009).

[26] A.D. King, *Personal Communication*, (2010).

[27] J. Krausz "Démonstration nouvelle d'un théorème de Whitney sur les réseaux", Mat. Fix. Lapok 50, 75-85, (1943).

[28] X. Li, and W. Zang "A Combinatorial Algorithm for Minimum Weighted Colorings of Claw-Free Perfect Graphs", Journal of Combinatorial Optimization 9, 331-347, (2005).

[29] L. Lovász "Normal hypergraphs and the perfect graph conjecture", Discrete Mathematics 2, 253-267, (1972).

[30] F. Maffray, and B. Reed "A description of claw-free perfect graphs", Journal of Combinatorial Theory, Series B 75, 134-156, (1999).

[31] G.J. Minty "On maximal independent sets of vertices in claw-free graphs", Journal of Combinatorial Theory, Series B 28, 284-304, (1980).

[32] D. Nakamura, and A. Tamura "A revision of Minty's algorithm for finding a maximum weighted stable set of a claw-free graph", Journal of the Operations Research Society of Japan 44, 194-204, (2001).

[33] P. Nobili, and A. Sassano "A reduction algorithm for the weighted stable set problem in claw-free graphs", In L. Adacher, M. Flamini, G. Leo, G. Nicosia, A. Pacifici, and V. Piccialli, editors, *Proc. 10th CTW*, 223-226, (2011).

[34] G. Oriolo, U. Pietropaoli, and G. Stauffer "A new algorithm for the maximum weighted stable set problem in claw-free graphs" In A. Lodi, A. Panconesi and G. Rinaldi, editors, Proceedings Thirteenth IPCO Conference, 77-96, (2008).

[35] G. Oriolo, U. Pietropaoli, and G. Stauffer "On the Recognition of Fuzzy Circular Interval Graphs", *Submitted Manuscript*, (2010).

[36] K.R. Parthasarathy, and G. Ravindra "The strong perfect graph conjecture is true for $K_{1,3}$-free graphs", Journal of Combinatorial Theory, Series B 21, 212-223,(1976).

[37] J. Petersen "Die Theorie der regulären graphs", Acta Mathematica 15, 193-220, (1891).

[38] U. Pietropaoli "Some classical combinatorial problems on circulant and claw-free graphs" PhD Thesis, Università di Roma La Sapienza, 2008. An extended abstract appeared in *4OR* 7(3), 297-300, (2009).

[39] N. Sbihi, "Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile", Discrete Mathematics 29, 53-76, (1980).

[40] A. Schrijver "Theory of Linear and Integer Programming", Wiley, New York, (1986).

[41] A. Schrijver "Combinatorial optimization. Polyhedra and efficiency (3 volumes)", Algorithms and Combinatorics 24. Berlin: Springer, (2003).

[42] R.E. Tarjan "Depth first search and linear graph algorithms", SIAM Journal Computing 1(2), 146-160, (1972).

[43] L.E. Trotter "Line perfect graphs", Mathematical Programming 12, 255-259, (1977).

[44] D.B. West "Introduction to graph theory", Pearson Education, (2001).