Bidimensionality and Graph Decompositions

Athanassios Koutsonas

$\mu \prod \lambda \forall$

March 2008

Preface

According to Isokrates, a great pedagogue of the classical era, truly educated is not he who bears knowledge and wisdom, but he who as a member of a society can offer valuable help to the community. Living in the same city where the rhetor was teaching, I have often found it difficult to maintain the motivation for studying pure mathematics; at times appearing similar to building castles on the sand.

Nevertheless, it was once again Isokrates, who believed that knowledge can be the only saviour of humanity. In modern times, the intent of knowledge is rarely knowledge itself as a personality molding process. On the contrary it has become merely the means for lesser purposes. On that account I have regarded it as vital not to sacrifice the mental tuition theoretical mathematics have to offer.

Reflecting on the time of working on this study, I realize that this master thesis – as everything else in my life so far – has been subject to my complexities, spontaneities, passions, tempers, devotions, faults, and of course to coincidences, weather conditions, kilometric distances, time schedules.

Therefore, I feel the need to express my sincerely gratitude to my supervisor Dimitrios M. Thilikos (Department of Mathematics, UOA), not only for introducing me to the beautiful world of mathematical research during his course of parameterized complexity and algorithms in the summer semester of 2005-06, and subsequently for the endless hours of coordination and guidance he devoted to me, but also for putting up with my idiosyncrasies.

Further, I would like to warmly thank Reinhard Diestel (Department of Mathematics, UHH) for his hospitality during my internship in the winter semester of 2006-07; his graph theory courses have been fascinating and most inspiring.

I also take the opportunity to express my deepest appreciation and to personally thank everyone involved in the program of MPLA, starting with the chairman of the program C. Dimitracopoulos and continuing with all my professors (in alphabetical order), I. Emiris, S. Kolliopoulos, E. Koutsoupias, Y. Moschovakis, D. Thilikos and E. Zachos.

In particular, I would like to thank the members of my committee: S. Kolliopoulos (Department of Informatics, UOA) and E. Zachos (School of Electrical & Computer Engineering, NTUA).

Finally, I want to thank my family, all my friends, co-students, and explicitly Kınay and Stephan for making it possible for me to feel like home, more than 15 degrees of latitude away from my own.

Athens, 14 March 2008.

Contents

Chapter 1

Introduction

Algorithms and complexity theory, these are terms any student of a mathematic or computer science department will often encounter, residing in harmony in one phrase. It will only require a longer period of time to actually comprehend, that much as the day and the night, the joy of the one relies on the absence of the other.

Under the light of the conjecture that P is not equal to NP, the class of interesting problems tends to be exactly the one consisting of NP-hard problems. Indeed usually, an interesting problem is difficult and a difficult problem is interesting.

And although complexity theory suggests that NP-hard problems should be rather left alone, even the next guy – blessed with the unawareness of prominent mathematical principles – will come across and successfully overcome more than one such problems in his everyday life.

The missing link can be interpreted in various ways. Accordingly, different schools of algorithm design have been developed: approximation algorithms, randomized algorithms, probabilistic algorithms, only to name a few. Arguments portrayed include "optimality is not an issue, good enough is good enough", or "that could very well be a solution to the problem".

A sensible approach is based on the observation, that most of the times only a limited sized instance is one of practical interest. As example, when asking how many guards are needed to prevent any acts of smuggling, one can rarely employ arbitrarily many guards. By this consideration, roughly speaking, emerges the theory of *fixed-parameter complexity and algorithms*, that has been developed during the last two decades [22, 23, 24, 21, 1].

Given an alphabet Σ, a parametrization of Σ^* is any computable in polynomial time mapping $\kappa : \Sigma \to \mathbb{N}$. A parameterized problem associated with Σ, is a pair (L, κ), where $L \subseteq \Sigma^*$ and κ a parametrization of Σ^*. An algorithm is called a *fixed parameter* algorithm, if there exists a countable function $f : \mathbb{N} \to \mathbb{N}$, such that for any $x \in \Sigma^*$ the algorithm replies in at most $f(k(x)) \cdot |x|^{O(1)}$ steps. A parameterized problem (L, κ) is *fixed parameter tractable* and, thus, contained in the parameterized complexity class FPT, if there is a fixed parameter algorithm on κ deciding L (for more on parameterized complexity see the surveys [29, 30, 27, 26, 25, 28], see also books [20, 33, 43]).

In algorithm theory, the means to formalize and approach most computational problems are provided by graph theory. A *graph* G is a pair of sets (V, E), the vertices and edges respectively, such that $E \subseteq [V]^2$. Having said that, the input of a parameterized problem can be expressed as (G, k). The parameter k, here, can be the size of a set of vertices incident to all edges, for example.

In this work, we focus on the planar versions of the problems considered, i.e. when the input graph of the problem is planar, or in other words it can be embedded on the sphere $\mathbb{S}_0$, so that its vertices and edges are pairwise distinct. We stress however that the same techniques can be implemented in designing algorithms handling wider classes of graphs, such as bounded genus graphs [11], apex-minor-free graphs [10] and H-minor-free graphs [11].

We study how efficient algorithms for a range of graph theoretical problems can be designed. In particular, we examine an established technique delivering sub-exponential parameterized algorithms for planar graphs, which much relies on *graph decompositions* (see Chapter 3). Introduced by Robertson & Seymour in their monumental proof of the Graph Minor Theorem ([]), graph decompositions provide a powerful tool for the implementation of algorithms. We analyze different ways of decomposing a graph, and how this allows us to employ a common strategy for addressing intractable problems, namely dynamic programming.

In fact, these techniques are steamed by the relatively new theory of *bidimensionality* (see Chapter 4) developed in [14, 11, 12, 10]. We survey the features of this theory, emphasizing on its generic applicability, which manages to encapsulate the basic structural properties of different graph parameters, delivering thus combinatorial bounds associated with a class of problems. Utilizing these, we can deliver algorithms of the desired complexity.

On this account, we proceed to a “tailor made“method, which requires a more precise understanding of the particular characteristics of the problem in question. Doing so, we prove deeper combinatorial results, which we use for the improvement of the algorithmic analysis of the parameterized version of three widely studied combinatorial problems on planar graphs. The first is the planar version of FEEDBACK VERTEX SET that asks, whether a planar graphcontains at most k vertices meeting all its cycles. The second is the FACE COVER that asks, whether all vertices of a plane graph G lie on the boundary of at most k of G. The last is the CYCLE PACKING that asks, whether a planar graph contains at least k disjoint cycles.

Historically, the FEEDBACK VERTEX SET, as well as its directed version, are one of the most studied *NP*-complete problems (for NP-completeness see [37]), mainly due to their numerous applications (see [32]). A wide range of algorithmic results on FEEDBACK VERTEX SET have been proposed including approximation algorithms [9, 39, 38], exact algorithms [34] and heuristics [42].

We consider the parameterized version of the three problems on planar graphs, namely p-PLANAR FEEDBACK VERTEX SET, p-FACE COVER and p-PLANAR CYCLE PACKING, where the integer k in the definition of the previously described problems is fixed as the parameter. These problems are solvable by subexponential FPT-algorithms ([41]), i.e. algorithms running in $O(2^{o(k)} \cdot n^{O(1)})$ steps (here, and generally, we denote by n the size of the input graph). In addition, Fernau and Juedes proved in [31] that FACE COVER can be solved in $O(2^{24.551\sqrt{k}} \cdot n)$ steps.

Following the approach outlined above, we prove that p-PLANAR FEEDBACK VERTEX SET, p-FACE COVER and p-PLANAR CYCLE PACKING can be solved in $O(2^{15.11 \cdot \sqrt{k}} + n^{O(1)})$, $O(2^{10.1 \cdot \sqrt{k}} + n^{O(1)})$ and $O(2^{26.347 \cdot \sqrt{k}} + n^{O(1)})$ steps, respectively. To our knowledge, these are the fastest, so far, algorithms for the mentioned problems.

Two corner stones of our proof are the use of *hypergraphs*, a natural extension of the notion of a graph where a hyperedge is now a non-empty set of vertices, and *plane duality*, a fascinating topic of the planar graph theory involving combinatorial and topological characteristics of the surface of the sphere $\mathbb{S}_0$ minus the embedded graph. It is their combination, which enables us to unify the analysis of both problems studied, by exploiting a duality relation between them.

Hypergraphs and plane duality, as well as other basic notions and proofs of simple propositions laying the scenery for the advanced material, are the subject of the following Chapter. Next, in Chapter 3 we gain an insight on graph decompositions; in particular we emphasize in *sphere-cut decompositions*, which hold an important role to several of our proofs. The theory of bidimensionality is analyzed in Chapter 4. Consequently, Chapter 5 is dedicated to the thorough analysis of the structure of face covers in planar graphs, which leads to the proof of our main result and to its reflection on the structure of feedback vertex sets in planar graphs, two combinatorial bounds of independent interest. Finally, in Chapter 6, we conclude the algorithmic consequences of our results, and discuss the interesting open problems emerged from this work.

Chapter 2

Basic Notions

Although most of the graph theoretical tools and techniques discussed in this study, affect wider classes of graphs, we focus our interest in plane graphs and hypergraphs. Dealing with planarity, our proofs share combinatorial and topological aspects. In this first chapter, we introduce the combinatorial notions of both a graph and a hypergraph, and get familiar with basic features. We, then, outline some simple topological facts in brief, allowing us to define plane embedings of graphs on the sphere.

Once this is done, we can proceed in presenting the concept of plane duality, an elegant topic on its own, which will hold an important role throughout the study. Complementing the selection of basic tools in our arsenal, we examine the radial and the medial graph, analyzing the way duality reflects upon them.

2.1 Graphs and Hypergraphs

A *graph* is a pair $G = (V, E)$, where V is a finite set, and E a set of subsets of V, each of which has exactly to elements. We call the elements of V *vertices* and the elements of E *edges*. If we relax the restriction above, so that an element of E is a not empty subset of the finite set V, than the pair $H = (V, E)$ defines a hypergraph; the elements of E are called, in that case, hyperedges.

The vertex set of a graph G is denoted as $V(G)$, its edge set as $E(G)$. Likewise for a hypergraph H, we have $V(H)$ and $E(H)$, respectively. The cardinality of the vertex set is the *size* of the (hyper-)graph, also denoted as

$|G|$ (and $|H|$).

Note, that a graph is a hypergraph, but the opposite is not necessarily true. For simplicity, we sometimes refer to graphs, including under the same notation hypergraphs as well, when it is clear why we are allowed to do so. Occasionally, when wishing to further stress that we exclude hypergraphs, we can use the terms *plain graph* and *trivial edge*.

With this in mind, we say that a vertex v is an *incident* to an edge e, if $v \in e$; then e lies on v and vice versa. The *degree* of a vertex is the number of edges this vertex lies on. Two vertices are *adjacent* or *neighbors*, if there is an edge e lying on both vertices. The *arity* of a hyperedge is the number of the vertices lying on the hyperedge. Trivial edges have, thus, arity equal to two. Vertices lying on a (hyper-)edge, are called its *endvertices*, and the edge *joins* its endvertices.

Two hypergraphs (or plain graphs) H_1, H_2 are *isomorphic*, if there is a bijection $\sigma : V(H_1) \to V(H_2)$, such that a non-empty subset e of $V(H_1)$ is a hyperedge in H_1, if and only if the set $\{\sigma(v) : v \in e\}$ is a hyperedge in H_2. We write, then, $H_1 \simeq H_2$.

Let $H = (V, E)$ be a hypergraph. If $V' \subseteq V$ and $E' \subseteq E$, then $H' = (V', E')$ is a *subgraph* of H, and we write $H' \subseteq H$. If in addition it holds that, if e is an edge in $E \setminus E'$, then there exists a vertex v incident to e with $v \in V \setminus V'$, the subgraph is called *induced*, and we write $H' = H[V']$. We also say, that V' induces the subgraph H' in H.

For the rest of the paragraph, we consider only plain graphs. For any integer $r \geq 1$, the graph $P_r = (\{v_1, \dots, v_{r+1}\}, \{\{v_1, v_2\}, \dots, \{v_r, v_{r+1}\}\})$ is a *path*, and vertices v_1, v_r are its *ends*, *linked* by the path. For $r \geq 3$, the graph $C_r = (\{v_1, \dots, v_r\}, \{\{v_1, v_2\}, \dots, \{v_{r-1}, v_r\}, \{v_r, v_1\}\})$ is a *cycle* of *length* equal to r. For $r \geq 1$, the graph on r vertices is called *complete* and denoted K_r, if all its vertices are pairwise adjacent. A *forest* is a *acyclic* graph, namely containing no cycles.

A non-empty graph G is called *connected*, if its vertices are pairwise linked by some path in G. Furthermore, G is k-*connected* for $k \geq 2$, if $|G| > k$ and G[V(G)-X] is connected, for every set $X \subseteq V(G)$ with $|X| < k$. A maximal connected subgraph of G is called a *component* of G. If $G[V(G) - S]$ for the set $S \subseteq V(G)$ is no more connected, S is called *separator*. If a single vertex is a separator, it is also called *cut-vertex*. Each component of a forest is called a *tree*; its vertices of degree one are its *leaves*.

Let G be a graph or a hypergraph. We say that a (hyper-)edge e of G is *contracted* into a new vertex v_e, when e and its endvertices are replaced by

v_e, which in turn becomes incident to all the (hyper-)edges, the endvertices of e were incident before the contraction. If a (hyper-)graph G' can occur from a subgraph of G by a series of (hyper-)edge contractions, we call G' a *minor* of G, and we write $G' \preceq G$. If a (hyper-)graph G'' can occur directly by a series of (hyper-)edge contractions of G, we write $G'' \preceq_c G$. (for more on basic notation see [15])

2.2 Planarity

Let $\mathbb{S}_0$ be a *sphere*. $\Delta \subseteq \mathbb{S}_0$ is an *open disc* if it is homeomorphic to $\{(x, y) : x^2 + y^2 < 1\}$. For a $\Delta \subseteq \mathbb{S}_0$, we call *closed disk* and denote as $\overline{\Delta}$ the *closure* of Δ; the *boundary* of Δ is $\widehat{\Delta} = \overline{\Delta} \cap \overline{\mathbb{S}_0 - \Delta}$.

A simply-closed curve or *Jordan curve* is the open subset of the sphere homeomorphic to the unit circle S^1. By the known theorem, a Jordan curve is the boundary of exactly two open discs.

An *arc* is the closed subset of $\mathbb{S}_0$ homeomorphic to the closed unit interval $[0, 1]$. The images of 0 and 1 under such a homeomorphism are the *endpoints* of the arc, which *links* them. Let γ be an arc with endpoints x, y; then its *interior* is the set $\mathring{\gamma} = \gamma \setminus \{x, y\}$. Let now $O \subseteq \mathbb{S}_0$ be an open set. Being linked by an arc in O defines an equivalence relation in O. The corresponding equivalence classes are the *regions* of O and are again open.

Let us define a *flake* A_ρ as homeomorphic to the closed disc, minus ρ (finite) points on its boundary (see also Figure 1.1):

$$A_\rho = \{(x, y) : x^2 + y^2 \leq 1\} - \{(\sin\frac{2k\pi}{\rho}, \cos\frac{2k\pi}{\rho}) : k = 0, \ldots, \rho - 1\}$$

We are now ready to give a definition of a plane hypergraph:

Definition 2.2.1. A *plane hypergraph* G is a pair $\{V, E\}$ of finite sets, (where $V = V(G)$ the vertex set and $E = E(G)$ the edge set) with the following properties:

1. $V \subseteq \mathbb{S}_0$ is a finite set of pairwise distinct points.
2. $E \subseteq \mathbb{S}_0$.
3. $\forall e \in E$, e is homeomorphic to A_ρ, for some $\rho \in \{2, \ldots, |V|\}$.

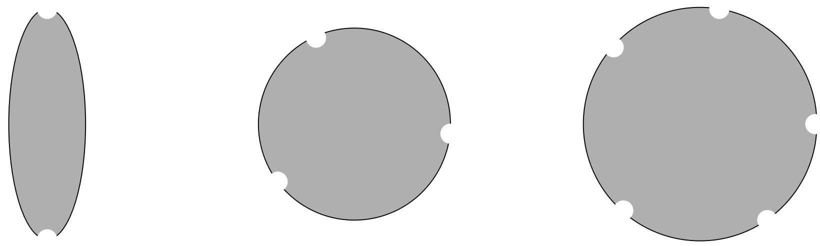

Figure 2.1: Hyperedges (or flakes) for $\rho = 2, 3$ and 5.

4. $\forall e \in E,\ \mathbf{bor}^*(e) \subseteq V$, where $\mathbf{bor}^*(e) = \mathbf{bor}(e) - e$.

5. $\forall e_1, e_2 \in E : e_1 \cap e_2 = \emptyset$.

6. $\forall v \in V, e \in E : v \nsubseteq e$.

A trivial edge, as a hyperedge of arity two, is homeomorphic to a flake A_ρ with $\rho = 2$. Hence, the definition above contains the definition of a plane graph. In addition, note that for a flake A_ρ with $\rho = 2$, the surface $\mathbb{S}_0 - A_\rho$ is homeomorphic to the surface $\mathbb{S}_0 - \mathring{\gamma}$ for an arc γ; this allows us to draw a trivial edge as the interior of an arc, following thus the common convention.

For every plane hypergraph G, the set $\mathbb{S}_0 \backslash G$ is open; its regions are the faces of G. We denote the set of the faces by $F(G)$. We say for a hyperedge e that it is incident to a face f, if $e \cap \bar{f} \neq \emptyset$ and for a vertex v if $v \subseteq \bar{f}$. We call a face f *degenerate*, *triangle* or *square* if its boundary contains exactly two, three or four vertices, respectively.

An embedding of a graph is a drawing of it on the sphere $\mathbb{S}_0$; graphs having a plane embedding are called *planar*. We will not be strict in distinguishing the terms “plane”and “planar”, when it is clear if we refer to topological or combinatorial properties of a graph.

Let G be a plane graph and f be a face of G. We denote as *face tiling* the operation, where a hyperedge e_f is added in the face f, i.e. $e_f \subseteq f$, with endvertices all vertices on the boundary of f. A graph G *generates* a hypergraph H, whenever H is obtained from G by tiling faces and deleting trivial edges that lie on the boundary of a already tiled face.

Closing the paragraph, let us stress that face tiling, as well as edge deleting or contracting does not harm the planarity; therefore graphs generated by planar graphs and minors of planar graphs are also planar.

2.3 Duality

Let us consider a plane graph G drawn on the sphere $\mathbb{S}_0$. As mentioned above, $\mathbb{S}_0 \backslash G$ is a division of the sphere surface into a finite set of regions, the faces of G. Suppose we aim to inspect the properties of the relative position of these faces. We can start drawing a graph, placing a vertex to represent each face of G and joining them by edges to include the information that a face shares a common border with another one. Continuing this, has as result a new graph, the *dual graph* of G denoted as G^*. Not surprisingly, G^* is also plane and its faces correspond to the vertices of the original graph. To make this formal:

Definition 2.3.1. Let $G = (V, E)$ be a connected plane graph and $G^* = (V^*, E^*)$ be its *dual*. Let $F := F(G)$ and $F^* = F(G^*)$ be the sets of the faces of the these two graphs. Then there exist bijections:

$$v^* : F \to V^* \qquad e^* : E \to E^* \qquad f^* : V \to F^*$$

such that following conditions are satisfied:

(i) $v^*(f) \in f$ for all $f \in F(G)$ and $v \in f^*(v)$ for all $v \in V(G)$

(ii) $\forall e \in E(G) : bor^*(e^*(e)) = \{v_o^* \in V^* : \overline{f}_o \cap e \neq \emptyset\}$, where $f_o = [v^*]^{-1}(v^*) \in F(G)$

(iii) if $e \cap e^*(e) \neq \emptyset$, then $e^* = e^*(e)$ for all $e \in E(G)$, $e^* \in E^*(G)$

Not all plane hypergraphs have plane duals according to the given definition of planarity. However, hypergraphs generated by 3-connected graphs always do. In the whole study we are interested only in this kind of hypergraphs, and therefore we can extend the features of duality to these special hypergraphs.

If we consider the dual graph of G^*, then by this definition we end up with the original graph $G = [G^*]^*$ (which explains the naming "dual") and so we can refer to both G and G^* as duals.

Trying to get an insight of the definition, the first condition assures that a vertex corresponding to a face of his dual graph, actually lies in that face. The second condition states that for every edge e of G, its dual edge e^* in G^* has endvertices the dual of the faces of G on whose boundary the edge e lies. By the last condition, edges that intersect on the sphere are bounded to be dual of its other. In addition, one can assume that this intersection is homeomorphic to a closed disc.

2.4 Radial and Medial Graph

Two important notions in the theory of planar graphs, which turn also to be powerful tools in argumentation, are the *radial* and the *medial* of a graph (introduced in [45] and [46], respectively). They offer a thorough display of the properties of a graph, revealing, often, the profound characteristics of its structure. Also, as we will show, the radial and the medial of a graph interplay with duality in fascinating ways, due exactly to their structural nature.

At first, given a graph or a hypergraph we define its radial, a bipartite plain graph on the vertices of the graph given and on new vertices lying in the faces of the graph, and whose (trivial) edges join two vertices of the radial, representing a face and a vertex on its boundary in the given graph. More precise:

Definition 2.4.1. Let G be a connected plane hypergraph. We call the bipartite plain graph (V_R, E_R) with bipartition $V_R = \{V(G), V^*\}$ a *radial graph* of G, and denote it $R_G = R(G)$, if there is a bijection $v^* : F(G) \to V^*$ satisfying the following conditions:

(i) $v^*(f) \in f$ for all $f \in F(G)$;

(ii) $e' \cap e = \emptyset$ for all $e' \in E_R$ and $e \in E(G)$;

(iii) $vv^*(f) \in E_R$ if and only if v lies on the boundary of f in G.

It is easy to see, that every edge of the graph G is mapped to a face of R_G, forming thus a bijection between $E(G)$ and $F(R_G)$. Let us now consider the radial of the dual of G. Clearly, both R_G and R_{G^*} have the same vertex set, namely $V(G) \cup V(G^*)$. Moreover, two vertices v_1, v_2 in R_{G^*} are joined, if v_1 in G^* lies on the corresponding face f_2 of G^*, meaning, if the corresponding face f_1 in G has the vertex v_2 of G on its boundary, or in other words if v_1, v_2 are joined in R_G, which explains the following statement:

Proposition 2.4.2. *Dual graphs have isomorphic radial graphs:* $R_G \simeq R_{G^*}$.

Let G be a 2-connected plain graph. Then, all faces of its radial R_G are squares and R_G itself is 3-connected. We define the *tiled radial graph* $\tilde{R}_G$ as the plane hypergraph generated from R, by tiling each face and removing all the trivial edges. Note that $\tilde{R}_G$ has only degenerate faces and that its hyperedges correspond to the edges of G.

Before being able to proceed to a definition of the medial, we need an intermediate graph called incidence graph. The definitions of both graphs, which are again of trivial edges only, are as follows:

Definition 2.4.3. Let G be a connected hypergraph, then $I(G)$, the *incidence graph* of G, is the simple bipartite plain graph with vertex set $V(G) \cup E(G)$, in which $v \in V(G)$ is adjacent to $e \in E(G)$ if and only if v is an end of e in G.

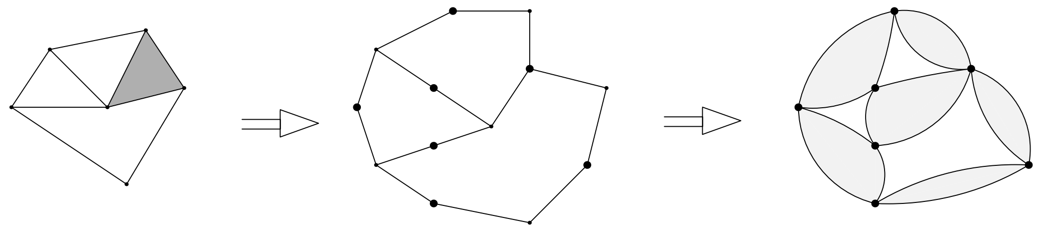

Figure 2.2: A hypergraph with its incidence and medial graph.

Definition 2.4.4. Take a drawing of $I(G)$ in a sphere. We define the *medial graph* $M_G = M(G)$, as a graph with vertex set $E(G)$ and circuits C_v ($v \in V(G)$), with the following properties:

- the circuits C_v are mutually edge-disjoint and have union M_G,

if $x_1, ..., x_t$ are the neighbors of each $v \in V(G)$ in $I(G)$, enumerated according to the cyclic order of the edges $vx_l, ..., vx_t$ in the drawing of $I(G)$, then

- C_v has vertex set $\{x_1, ..., x_t\}$ and $x_{(i-1)}$ is adjacent to x_i ($1 <= i <= t$), where x_0 means x_t.

The construction of the medial M_G of a hypergraph G can be described in an alternate manner: given G, first we stretch each vertex to become a face having one vertex on its boundary to be joined to each edge the original vertex of G was incident to, then we contract all edges to become the vertices of M_G. So, we have bijections between $V(G) \cup F(G)$ and $F(M_G)$, and between $E(G)$ and $V(M_G)$.

By this second description, it gets obvious that the medial of a graph G has to "types"of faces, those emanated from the vertices of G and those emanated from the faces of G. Moreover, if we paint them using two colors, we will see that there are no two faces of the same color sharing an edge. In other words the set of the faces of the medial is bipartite: $F(M_G) = (\overline{V(G)}, \overline{F(G)})$. If a face of the first type is lying next to one of the second, is determined by the fact, that the origin vertex in G lies on the boundary of the origin face in G or not. It is becoming apparent that:

Proposition 2.4.5. *Given a graph G, its radial and medial are dual graphs: $R_G \simeq M_G^*$.*

The Propositions 2.4.2 and 2.4.5 directly implicate the next one, since given a graph G, it suffices to take the radial of both G and G^*, resulting R_G and R_{G^*}, then take the dual of both of them, resulting M_G and M_{G^*}, and as R_G and R_{G^*} where homeomorphic, we have again:

Proposition 2.4.6. *Dual graphs have isomorphic medial graphs: $M_G \simeq M_{G^*}$.*

It is not difficult, besides, to see why this must be true: both M_G and M_{G^*} have the same vertex set, the image of $E(G)$, the same face set, the image of $V(G) \cup F(G)$ and an edge joining two vertices, iff two edges of $E(G)$, or equivalently, iff the corresponding edges of $E(G^*)$ are adjacent. Remembering our two colors, one would observe that M_G and M_{G^*} look the same, but with complement colors on their faces: where a face is of color one in M_G, is of color two in M_{G^*} and likewise for the other combination.

With a given hypergraph G, taking its dual G^*, taking its radial R_G and taking its medial M_G, can also be viewed as a transformation between two hypergraphs G_1 and G_2, described by the projection of $\{V_1, E_1, F_1\}$ into $\{V_2, E_2, F_2\}$. These transformations would then be described by the as:

$$\begin{aligned}
Dual: &\quad \{V_1, E_1, F_1\} &\rightarrow&\quad \{F_2, E_2, V_2\}\\
Radial: &\quad \{V_1, E_1, F_1\} &\rightarrow&\quad \{V_2, F_2, V_2\}\\
Medial: &\quad \{V_1, E_1, F_1\} &\rightarrow&\quad \{F_2, V_2, F_2\}
\end{aligned}$$

Of course, as one can clearly see, the radial and the medial graph are, by this alone, not well defined, lacking information about their edge set. As

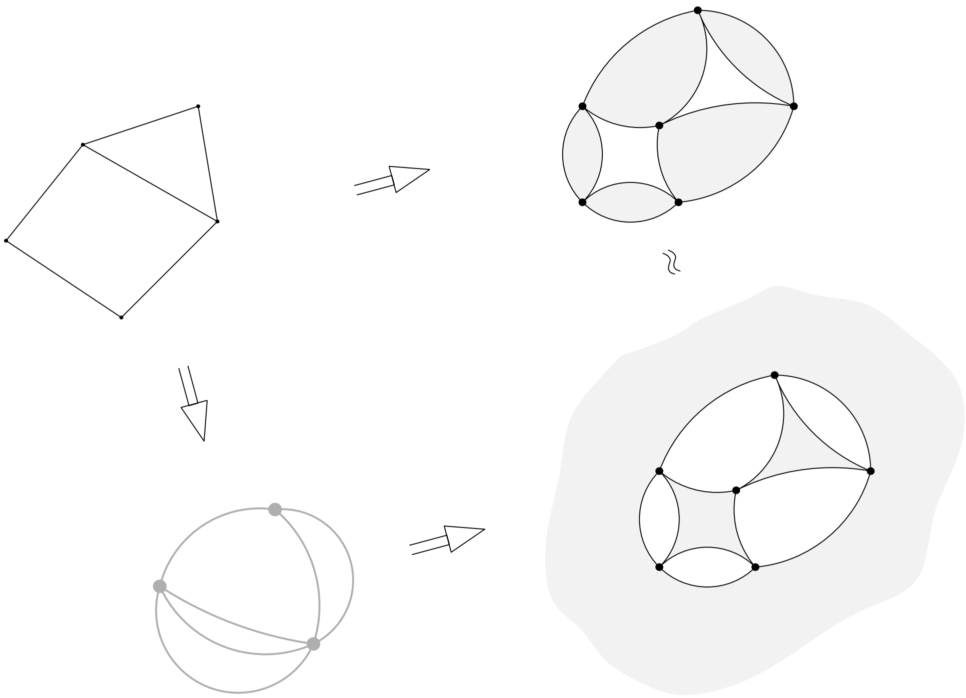

Figure 2.3: Dual graphs have homeomorphic medial graphs.

we know, in the first case an edge represents the fact that a vertex lies on the boundary of a face in G_1 and in the second the fact, that two edges are adjacent in G_1.

It is now very interesting to examine from this point of view, the above mentioned propositions, where a combination of two different projections can be considered. One can see as example, how applying two times the projection of duality to a graph, one receives at the end the same graph, or how the way the projection is defined in the cases of the radial and medial graph, enables them to be symmetric regarding the projection of duality firstly, each to itself and secondly, each to another.

Also, one can proceed further in defining other types of transformations; one of the most interesting amongst them is the case of the *complement graph* C_G, of a given hypergraph G, described as followed:

$$\textit{Complement}: \qquad \{V_1, E_1, F_1\} \qquad \rightarrow \qquad \{V_2, F_2, E_2\}$$

Using, as example, this transformation, one can define the incidence graph

$I(G)$ of a given graph G, as: $Incidence(G) = Radial[Complement(G)]$.

Chapter 3

Graph Decompositions

Introduced by Robertson, Seymour and Thomas in their series of papers as a tool for the ultimate objective of proving the Graph Minor Theorem, Graph Decompositions have been since, the steam for the development of numerous methods and proving techniques. In particular, in fields as the algorithm design for graph-based (and not only) problems, their implementation can be regarded as fundamental. In this chapter, we are going to describe three different approaches of how to decompose a graph, each with its own beneficial characteristics and finally prove a significant result, namely the existence of an optimal sphere-cut decomposition.

3.1 Tree Decompositions

In terms of algorithm design, one would only wish that all graphs were as simple as trees, enabling thus the use of powerful tools like dynamic programming. However, it is reasonable to expect, that such techniques can be implemented on graphs other than trees, as long as we can keep track of a structure to guide us through the graph. The idea is, to have this tree-like structure so we can move fast enough to reach smaller sections of the graph, in which it would not matter any more, if they had to be traversed exhaustedly. Not so far away, from the manner the street system of a city is organized, where a spine of runways connects the different districts with their smaller streets. The structure, which gives us this ability, is called *tree-decomposition* and defined as follows (see also [15]):

Definition 3.1.1. Let G be a graph, $\mathcal{T}$ a tree, and let $\mathcal{V} = (V_t)$, $t \in \mathcal{T}$ be a family of vertex sets $V_t \subseteq V(G)$ indexed by the vertices t of $\mathcal{T}$. The pair $(\mathcal{T}, \mathcal{V})$ is called a *tree-decomposition* of G, if it satisfies the following three conditions:

- $V(G) = \bigcup_{t \in \mathcal{T}} V_t$
- for every edge $e \in G$, there exists a $t \in \mathcal{T}$, such that both ends of e lie in V_t
- $V_{t_1} \cap V_{t_3} \subseteq V_{t_2}$, whenever $t_1, t_2, t_3 \in \mathcal{T}$ satisfy $t_2 \in t_1 \mathcal{T} t_3$.

So, the tree of the decomposition is defined upon subsets of the vertex set of the graph, informally referred to as 'bags'. The conditions above assure, that the union of the subgraphs induced by these subsets covers the whole graph and that the tree follows the structure of the graph.

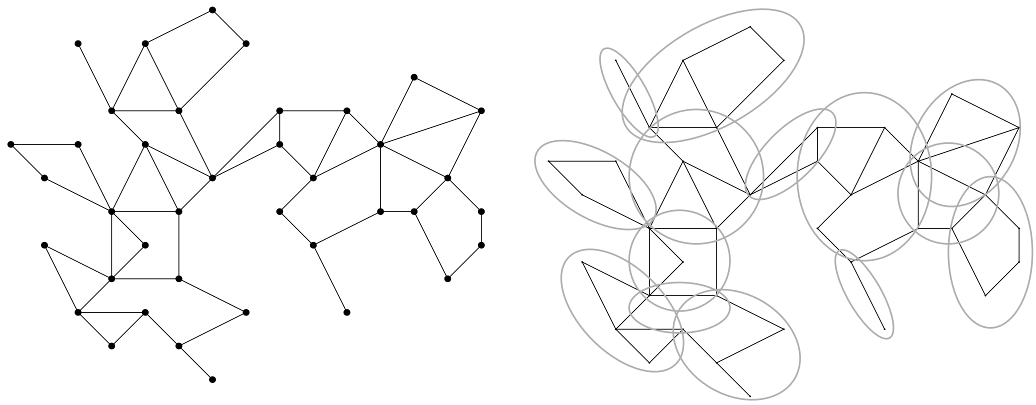

Figure 3.1: A plane graph and its decomposition in subgraphs.

Let G be a graph, and $(\mathcal{T}, \mathcal{V})$ a tree decomposition of G. Removing a vertex t of T, leaves a collection of at least two disconnected subtrees. The critical feature of tree decompositions is that, by the third condition, the vertices V_t in the bag of t are a separator in the graph; the components remaining after removing these vertices are exactly the subgraphs induced by the mentioned subtrees. This is exactly what allows for routines as dynamic programming to work ([4]).

To be precise, a stronger property of separation holds: Deleting any edge in the tree $\mathcal{T}$ of a decomposition of a graph G, leaves two disconnected subtrees. These induce two subgraphs, whose union is G. By the definition, their cut is a separator in G.

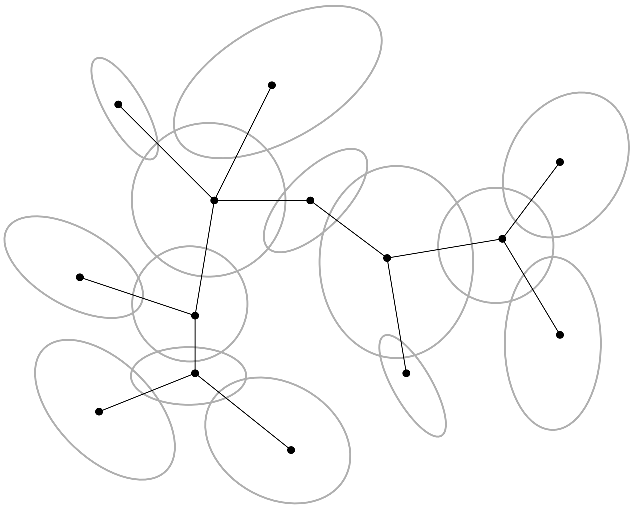

Figure 3.2: The Tree-Decomposition of the previous graph.

Naturally, one would wish to refine as much as possible the partition of the vertex set into subsets, as this resolves to a structure better resembling a tree; recall that to associate the whole vertex set of a graph to a single vertex of the tree is still an acceptable decomposition. In these terms, we can measure the effectiveness of an decomposition by the number of the vertices of the heaviest subset corresponding to a vertex of the tree. Thus, we define its *tree-width*:

Definition 3.1.2. Let G be a graph and $(\mathcal{T}, \mathcal{V})$ be a tree-decomposition of G. The *width* of $(\mathcal{T}, \mathcal{V})$ is the number

$$\max\{|V_t| - 1 : t \in \mathcal{T}\}$$

and the *tree-width* $tw(G)$ of G is defined as the minimum width over all tree-decompositions of G.

The sole purpose of subtracting one in this definition, is because of the reasonable request, of forests to have tree-width equal to 1. It is easy to find one that satisfies this. Placing the endvertices of each edge of the tree in a different bag, leads to an acceptable tree-decomposition, with all its bags containing exactly two vertices, implicating that its width is equal to 1.

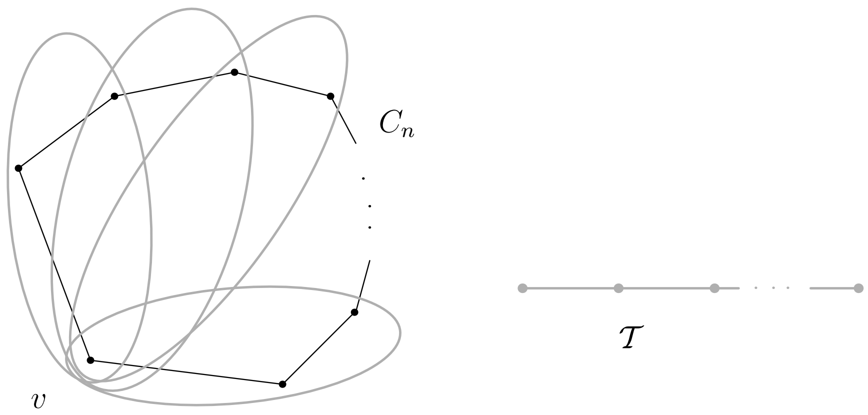

Figure 3.3: A tree-decomposition of C_n.

Notice that tree-width is a property of a graph and whether one can find a tree-decomposition of this width, for the graph in question, is task of its own. Any tree-decomposition of a given graph of width equal to the tree-width of the graph, will be denoted as *optimal*.

Clearly, deleting or contracting an edge cannot increase the tree-width of a graph, implicating that the following property is true:

Proposition 3.1.3. *Let G, H be graphs. If $H \preceq G$, then $tw(H) \leq tw(G)$.*

We already saw, that the forests have tree-width equal to 1. Furthermore, these are the only graphs that do so, as the existence of a cycle in a graph forces it to have tree-width of at least 2. In fact, the graph C_n has tree-width equal to 2 for any integer $n \geq 3$:

Choosing a vertex v at random we form a vertex subset with v and the first and second vertex on its left. Next bag will include v, the second and third vertices on its left, and continuing like this until we pack the right neighbor of v, we have a tree-decomposition where the tree is a chain and each bag has three vertices; hence of width equal to 2.

On the other side, the complete graph K_n, has tree-width $n-1$, for $n \geq 2$, as it can be showed that a bag must contain the whole graph. This gives us a certificate of large tree-width – if we can find a complete subgraph in a graph, then we know the total tree-width is at least that large. Are there other certificates, to look for? The most typical is the grid:

Definition 3.1.4. The $m \times m$ *grid* is the graph on $\{1, 2, \ldots, m^2\}$ vertices $\{(i, j) : 1 \leq i, j \leq m\}$ with the edge set

$$\{(i, j)(i', j') : |i - i'| + |j - j'| = 1\}.$$

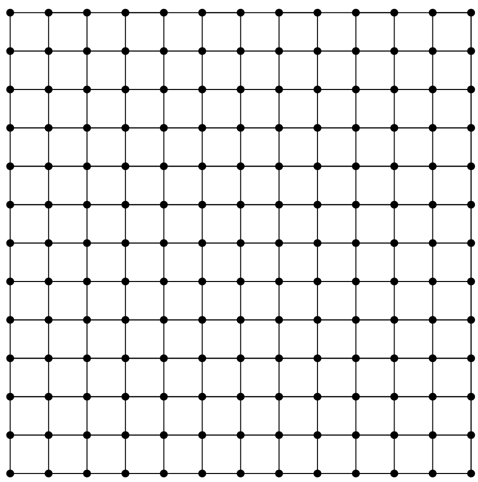

Figure 3.4: The 13×13-grid.

For $i \in \{1, 2, \ldots, m\}$ the vertex set $(i, j), j \in \{1, 2, \ldots, m\}$, is referred as the *ith row* and the vertex set $(j, i), j \in \{1, 2, \ldots, m\}$, is referred to as the *ith column* of the $m \times m$ grid. The vertices (i, j) of the $m \times m$ grid with $i \in \{1, m\}$ or $j \in \{1, m\}$ are called boundary vertices and the rest of the vertices are called non-boundary vertices.

With a careful look, one can show that the $m \times m$-grid has a tree-decomposition of width equal to m: Put the first row into a bag together with the first one vertex of the second row. The second bag contains the first row, except its first vertex and the first two vertices of the second row. Letting one out, putting one in and so on, again we have a decomposition where the tree is a chain, with all bags having $m+1$ vertices.

But even more interesting and a lot more difficult to prove, is the other direction, namely that this is the best we can do, which we will frame into the following proposition:

Proposition 3.1.5. *The $m \times m$ grid has tree-width equal to m, where m a positive integer.*

And with this stated, we leave the study of tree-decompositions, only to turn to its next to kin, branch-decompositions. (for more on tree-width see the survey [5])

3.2 Branch Decompositions

Similar to a tree decomposition, a branch decomposition of a graph consists of a tree and a relation between this tree and the graph. This time, it is subsets of the edge set of the graph that will be mapped to vertices of the tree. However, only leaves of the tree correspond to edges of the graph (see also [44]:

Definition 3.2.1. A *branch decomposition* of a graph G is a pair $(\mathcal{T}, \tau)$, where $\mathcal{T}$ is a tree whose vertices are either leaves or have degree three, and τ a bijection from the edge set of G to the set of the leaves of $\mathcal{T}$.

The function $\omega : E(\mathcal{T}) \to 2^{V(G)}$ of a branch-decomposition maps every edge e of $\mathcal{T}$ to a subset of vertices $\omega(e) \subseteq V(G)$ as follows: The set $\omega(e)$, called *middle set*, consists of all vertices $v \in V(G)$, such that there exist edges $f_1, f_2 \in E(G)$ with $v \in f_1 \cap f_2$, and such that the leaves $\tau(f_1), \tau(f_2)$ are in different components of $\mathcal{T} \setminus \{e\}$.

Definition 3.2.2. Given a graph G and a branch-decomposition $(\mathcal{T}, \tau)$, the width of $(\mathcal{T}, \tau)$ is equal to the number

$$\max\{|\omega(e)| : e \in E(\mathcal{T})\}.$$

We define the *branch-width* of the graph G, $bw(G)$, is the minimum width over all branch-decompositions of G.

As one would expect, the values $\mathbf{tw}(G)$ of the tree-width and $\mathbf{bw}(G)$ of the branch-width of any given graph G, can never diverge too much from each other. Specifically, they obey to the following relation:

$$\mathbf{bw}(G) \leq \mathbf{tw}(G) + 1 \leq \frac{3}{2}\,\mathbf{bw}(G),$$

and thus, whenever one of these parameters is bounded, so is the other.

Same as for tree decompositions, edge deleting or contracting does not increase the branchwidth of a graph, and thus:

Lemma 3.2.3. *Let G be a plane graph and let G' be a minor of G. Then* $\mathbf{bw}(G') \leq \mathbf{bw}(G)$*.*

Recall now the definition of the branch decomposition. Note that exactly as it is, stands for hypergraphs as well. Consequently, hypergraphs have also branch decompositions, and branchwidth is defined for hypergraphs. The following lemma, relates the branchwidth of a graph and of any hypergraph generated by it:

Lemma 3.2.4. *Let G be a plane graph and let H be a hypergraph generated by G. Then* $\mathbf{bw}(G) \leq \mathbf{bw}(H)$*.*

The last lemma is useful for gluing together branch decompositions of hypergraphs.

Lemma 3.2.5 ([35, Lemma 3.1])**.** *Let H_1 and H_2 be hypergraphs with one hyperedge in common, i.e. $V(H_1) \cap V(H_2) = e$ and $\{e\} = E(H_1) \cap E(H_2)$. Then, it holds that:* $\mathbf{bw}(H_1 \cup H_2) \leq \max\{\mathbf{bw}(H_1), \mathbf{bw}(H_2), |e|\}$*. Moreover, if every vertex $v \in f$ has degree ≥ 2 in at least one of the hypergraphs, (i.e. v is contained in at least two edges in $\mathcal{G}_1$ or in at least two edges in $\mathcal{G}_2$), then* $\mathbf{bw}(\mathcal{G}_1 \cup \mathcal{G}_2) = \max\{\mathbf{bw}(\mathcal{G}_1), \mathbf{bw}(\mathcal{G}_2)\}$*.*

Although computing the branchwidth of a general graph is NP-complete, when restricted to planar graphs, Seymour and Thomas proved it being in P, suggesting an $O(n^4)$ step algorithm [46] (this algorithm has been improved later to an $O(n^3)$ step algorithm [40]).

3.3 Carvings

A third way of decomposing a graph is attained by carvings (introduced in [46]. This approach is in fact more abstract, as it does not necessarily address a graph.

Definition 3.3.1. Let V be a finite set with $|V| \geq 2$. Two subsets $A, B \subseteq V$ *cross* if $A \cap B, A - B, B - A, V - (A \cup B)$ are all non-empty. A *carving* in V is a set $\mathscr{C}$ of subsets of V such that:

1. $\emptyset, V \notin \mathscr{C}$
2. no two members of $\mathscr{C}$ cross, and
3. $\mathscr{C}$ is maximal subject to 1. and 2.

A branch decomposition can be seen as a carving on the edge set of a graph. More precise, let V be a finite set with $|V| \geq 2$, let $\mathcal{T}$ be a tree in which every vertex has degree 1 or 3, and let τ be a bijection from V onto the set of leaves of $\mathcal{T}$. For each edge e of $\mathcal{T}$, let $\mathcal{T}_1(e), \mathcal{T}_2(e)$ be the two components of $\mathcal{T} \setminus e$ and let $\mathscr{C} = \{\{v \in V : \tau(v) \in V(\mathcal{T}_i(e))\} : e \in E(\mathcal{T}), i = 1, 2\}$. Then $\mathscr{C}$ is a carving in V. Conversely, every carving in V arises from some tree $\mathcal{T}$ and bijection τ in this way.

Definition 3.3.2. Let G be a graph. For $A \subseteq V(G)$, we denote by $\delta(A)$ the set of all edges with an end in A and an end in $V(G) - A$. If $|V(G)| \geq 2$ we define the *carving-width* of G to be the minimum, over all carvings $\mathscr{C}$ in $V(G)$, of the maximum, over all $A \in \mathscr{C}$, of $|\delta(A)|$.

We will prove the next lemma, associating the branchwidth of a graph with the carving-width of its medial:

Lemma 3.3.3. *Let G be a connected planar graph with $|E(G)| \geq 2$ and M_G its medial graph. Then, $\mathbf{bw}(G) \leq 1/2 \cdot \mathbf{cw}(M_G)$.*

Proof. Put $2m = \mathbf{cw}(M_G)$. We will show that $bw(G) \leq m$. By definition, there exists a carving $\mathscr{C}$ of the vertex set of M_G, such that $|\delta(A)| \leq 2m$ for any set A in $\mathscr{C}$. Due to the bijection between the vertex set of M_G and the edge set of G, $\mathscr{C}$ is also a carving on $E(G)$. Considering the observation that followed the Definition 3.3.1, the carving $\mathscr{C}$ on $E(G)$ yields a branch decomposition in G, let it be named $(\mathcal{T}, \tau)$.

Now let us calculate its width. Let e be an edge of $\mathcal{T}$. Its middle set $\omega(e)$ consists of all vertices incident to edges mapped to leaves in different components of $\mathcal{T}$. These vertices of G correspond, as we have seen, to faces of M_G not sharing any edges, which means that each face contributes at least two edges to $|\delta(A)|$, where $A \subseteq V(M_G)$ is the image of the edges of $E(G)$ mapped to leaves of one component of $\mathcal{T} \setminus \{e\}$. In other words, we have:

$$|\omega(e)| \leq \frac{1}{2} |\delta(A)| \leq m.$$

And since we chose e arbitrary, the same holds for the middle set of any edge of $\mathcal{T}$ and thus the width of $(\mathcal{T}, \tau)$ is at most m. This places an upper bound for the branch-width of G as well, concluding the proof. □

The opposite direction seems to be a lot more complicated to prove. A long proof involving *slopes* and *antipodalities* has been given by Seymour and Thomas (Theorem 7.2 [45, 46]). Hence, we have that the branchwidth of a graph is half the carvingwidth of its medial:

Theorem 3.3.4. *Let G be a connected planar hypergraph with $|E(G)| > 2$, and let M be the medial graph of G. Then $\mathbf{bw}(G) = 1/2 \cdot \mathbf{cw}(M_G)$.*

We are in particular interested, in a carving of a specific structure, such that any set contained in the carving induces a non-empty connected graph. Therefore we define:

Definition 3.3.5. If a graph G is connected, and $X, Y \subseteq V(G)$ are disjoint with union $V(G)$, and $G[X]$, $G[Y]$ are both non-null and connected, we call $\delta(X)$ a *bond* of G.

Definition 3.3.6. Let G be a connected graph. A carving $\mathscr{C}$ in $V(G)$ is a *bond carving* if $\delta(X)$ is a bond for all $X \in \mathscr{C}$.

The next theorem is one deep result, which assures us that given any graph, a bond carving of no greater width than the carving-width of the graph does always exist (Theorem (5.1) in [46]):

Theorem 3.3.7 ([46])**.** *Let G be a 2-connected graph with $|V(G)| \geq 2$ and carving-width $< k$. Then there is a bond carving $\mathscr{C}$ in $V(G)$ such that $|\delta(X)| < k$ for all $X \in \mathscr{C}$.*

3.4 Sphere-Cut Decompositions

We already commented that a branch decomposition is a carving on the edge set of a graph. If this carving is in addition a bond carving, then the associated branch decomposition bears special properties. We denote it as a sphere-cut decomposition (see also [18]):

Definition 3.4.1. A branch-decomposition $(\mathcal{T}, \tau)$ is called a *sphere-cut* decomposition, if for every hyperedge e of $\mathcal{T}$, there exists a noose O_e, such that:

- $G_i \subseteq \Delta_i \cup O_e$ for $i = 1, 2$, where G_i the subgraph induced by the vertices mapped to the leaves of the component $\mathcal{T}_i(e)$ of $\mathcal{T} \setminus e$ and Δ_i the open disc bounded by O_e,
- for every face f of G, $O_e \cap f$ is homeomorphic to the interior of exactly one arc on $\mathbb{S}_0$, linking two vertices of the boundary of f.

As Theorem 3.3.7 assures that there always exists a bond carving, we can expect that so does an optimal sphere-cut decomposition (one of minimum width):

Theorem 3.4.2. *Let G be a 2-connected planar hypergraph with $\mathbf{bw}(G) \leq k$ and $|E(G)| \geq 2$. Then, there exists a sphere-cut decomposition of G of width at most k.*

Proof. Let us consider the graph G and its medial graph M_G, both drawn on the $\mathbb{S}_0$-sphere. Note that, by definition, all edges of M_G are trivial. Furthermore, by Theorem 3.3.4 and Theorem 3.3.7, we know that M_G has a bond carving $\mathscr{C}$ of width at most $2k$.

Let $\mathcal{T}$ be the tree associated with the carving $\mathscr{C}$ and η the bijection between the leaves of $\mathcal{T}$ and the vertex set of M_G. For each edge $e \in \mathcal{T}$, let $\mathcal{T}_1(e), \mathcal{T}_2(e)$ be the two components of $\mathcal{T} \setminus e$ and $V_i = \eta(\text{"}\textit{leaves of } \mathcal{T}_i(e)\text{"})$ for $(i = 1, 2)$. We remind, that $V_1 \cup V_2 = V(M_G)$, $V_1 \cap V_2 = \emptyset$ and the induced subgraphs $M_G|V_1$, $M_G|V_2$ are both connected, since $\mathscr{C}$ is a bond carving.

Let $\bigcup_j \Delta_i^j$, for $i = 1, 2$ and $j \in \mathbb{N}$, be the union of open discs on $\mathbb{S}_0$, so that $M_G|V_i$ lies in $\bigcup_j \Delta_i^j$. Since the two subgraphs are connected, each union of the open discs is again an open disc, $\Delta_i' := \bigcup_j \Delta_i^j$, $(i = 1, 2)$. And because the subgraphs are induced and share no vertex in common, there is a jordan curve on $\mathbb{S}_0$ that bounds two discs Δ_1, Δ_2, so that $\Delta'_i \subseteq \Delta_i$ for $i = 1, 2$. We call this jordan curve a *sphere cut* related to e and denote it as Φ_e.

Recalling the drawing of M_G on $\mathbb{S}_0$, one can observe that Φ_e passes through no vertex (all vertices lie in some of the two open discs) and crosses exactly those edges of M_G, whose endvertices are mapped in different components of $\mathcal{T} \setminus e$. With no loss of generality, we can assume that it crosses these edges exactly once.

Furthermore, the intersection of Φ_e and each face of M_G it passes through, is a homeomorphic to an arc. To see this, contract all vertices of M_G that lie on the same disc bounded by Φ_e into two vertices; all faces, now, are clearly crossed by Φ_e exactly once. Note, also, that the number of these faces (and hence of their corresponding faces before the contraction) is equal to the number of the edges crossed by Φ_e.

Observe that $(\mathcal{T}, \eta)$ is a sphere-cut decomposition of G. First of all, recall that M_G has as vertex set the edge set of G; so now η naturally maps the leaves of $\mathcal{T}$ to the edges of G. Furthermore, that each vertex v of G corresponds to the circuit C_v together with the face it bounds in M_G, while the rest of the faces of M_G are mapped to the faces of G. Two faces of different "type"(see also Paragraph 2.4), can share no common edge, because no two circuits can neither.

Thus, the property A of a sphere-cut Φ_e, implicates that Φ_e crosses equal number of faces, say β, of each of the two "types", in an alternating manner and nothing but those. And due to the bijections mentioned above, the sphere-cut Φ_e for M_G yields a noose B_e for G, that passes through β vertices, exactly two boundary vertices of each of the β faces of G it crosses. And as Φ_e bounded two open discs, in which the vertices of M_G mapped to the leaves of the two different components of $\mathcal{T} \setminus e$ lay, same holds for the two open discs bounded by B_e and the edges of G. This concludes that $(\mathcal{T}, \eta)$ is a sphere-cut branch-decomposition of G.

Let us, finally, calculate its branch-width. For every edge e of $\mathcal{T}$ the edges that contribute to the calculation of the carving-width, are these that have one end in V_1 and one in V_2, i.e. these that are crossed by the sphere-cut Φ_e. As we already confirmed, the number of these edges is equal to the number of the faces crossed by Φ_e. And as showed, if Φ_e crosses 2β faces, then the noose B_e passes through β vertices. This is true for every edge e of $\mathcal{T}$ and, thus, the branch-width of $(\mathcal{T}, \eta)$ is half the carving-width of $\mathscr{C}$, i.e. equal to k, which concludes the proof. $\square$

Corollary 3.4.3. *For any planar hypergraph G (generated by a 3-connected graph), the branchwidth of G is equal to the branchwidth of its dual.*

Proof. Since the hypergraph G and its dual G^* have isomorphic medial graphs $M_G \simeq M_{G^*}$ (Proposition 2.4.6), applying Theorem 3.4.2 will deliver an optimal sphere-cut decomposition for each of the two hypergraphs, of equal branchwidth. □

In the case of plain graphs, the Corollary 3.4.3 holds for any graph that is not a forest.

Recall now the definition of the tiled radial $\tilde{R}_G$ of a graph G. With the use of sphere-cut decompositions, we can relate the branchwidth of the two graphs. It follows that the branchwidth of a graph is at least half of the branchwidth of its radial graph.

Lemma 3.4.4. *For any 2-connected plane graph G, it holds that $\mathbf{bw}(\tilde{R}_G) \leq 2 \cdot \mathbf{bw}(G)$.*

Proof. By Theorem 3.4.2, any plane graph G of $\mathbf{bw}(G) \leq k$ has a sphere-cut decomposition (T, μ) of width $\leq k$. By the definition of a sphere-cut decomposition, the middle set of e in (T, μ) is equal to $N_e \cap V(G)$ and thus $|N_e| \leq k$. Observe also that the noose N_e can be seen as a cycle C_e of the radial graph G_R of length twice the length of N_e.

Recall now that the definitions of R_G and $\tilde{R}_G$ implies the existence of a bijection $\rho : E(G) \to E(\tilde{R}_G)$ between the edges of G and the hyperedges of $\tilde{R}_G$. This permits us consider the branch decomposition (T, σ) of $\tilde{R}_G$ where $\sigma = \rho \circ \mu$ is the composition of the bijections μ and ρ. Observe that for any $e \in E(T)$, the middle set of e in (T, σ) consists of the vertex set of the cycle C_e. Therefore, (T, σ) of $\tilde{R}_G$ has width at most twice the width of (T, μ) and the lemma follows. □

Theorem 3.4.5. *Let G be a plane graph with $|E(G)| \geq 2$ and R_G its radial, then $\mathbf{bw}(R_G) \leq 2 \cdot \mathbf{bw}(G)$.*

Proof. If G is 2-connected, then by Lemma 3.4.4 its tiled radial $\tilde{R}_G$ is of branchwidth at most twice the branchwidth of G. Recall now that $\tilde{R}_G$ is generated by R_G, and therefore by Lemma 3.2.4 we have that $\mathbf{bw}(R_G) \leq \mathbf{bw}(\tilde{R}_G)$. Combining the two inequalities, we derive that $\mathbf{bw}(G) \leq 2 \cdot \mathbf{bw}(R_G)$ as wanted.

Assume that G is maximal not 2-connected. Since G has at least two edges, the branchwidth of G cannot be equal to zero. If $\mathbf{bw}(G) = 1$, it follows trivially by the definition of the radial graph, that $\mathbf{bw}(R_G) = 2$. Let us then assume that $\mathbf{bw}(G) \geq 2$.

Let v be the only cut vertex in G and G_1, G_2 the subgraphs joined by v. One of the two subgraphs is forced to have at least two edges, so by Lemma 3.2.5 it is $\mathbf{bw}(G) = max\{\mathbf{bw}(G_1), \mathbf{bw}(G_2)\}$. The two radial graphs R_{G_1} and R_{G_2} share exactly one edge, let it be e. Consider two optimal branch decompositions of R_{G_1} and R_{G_2}, join the leaves corresponding to e by an edge, subdivide it, and hang on this vertex the double edge e and e' associated with the cut vertex v. The resulting branch decomposition describes R_G and has width equal to the heaviest of R_{G_1} and R_{G_2}. Graphs G_1, G_2 are 2-connected, so we conclude again that $\mathbf{bw}(G) \leq 2 \cdot \mathbf{bw}(R_G)$. □

Chapter 4

Bidimensionality

The theory of bidimensionality, developed recently in the work of Demaine, Fomin, Hajiaghayi and Thilikos ([14, 11, 12, 10]), provides general techniques for designing efficient fixed-parameter algorithms and approximation algorithms for NP-hard graph problems in broad classes of graphs, namely all generalizations of planar graphs. Here, we are going to focus on its use in planar graphs. The theory can be applied to a series of well-known graph problems, such as vertex cover, feedback vertex set, face cover and dominating set, only to name a few. In the next paragraphs we proceed to a report of the main aspects of the bidimensionality theory.

4.1 General

We define parameters as an alternative view on optimization problems. A *parameter* $\mathbf{p}$ is any function mapping graphs to nonnegative integers. For a minimization (maximization) problem associated with $\mathbf{p}$, the decision problem asks, for a given graph G and nonnegative integer k, whether $\mathbf{p}(G) \leq k$ (respetively $\mathbf{p}(G) \geq k$). Many optimization problems can be phrased as such decision problems about a graph parameter. Let us see some examples:

(p_1) : "The maximum vertex degree of the given graph G"

(p_2) : "The tree-width of the given graph G"

(p_3) : "The minimum cardinality of a set of vertices in G, such that any vertex is in this set or adjacent to a vertex in it. "

We say that a parameter $\mathbf{p}$ is *closed under taking of minors*, if for every graph H, $H \preceq G$, implies that $\mathbf{p}(H) \leq \mathbf{p}(G)$. Similar, P is *closed under contractions*, if for every graph H, $H \preceq_c G$, implies that $\mathbf{p}(H) \leq \mathbf{p}(G)$. As we already have seen, the second follows directly from the first, but the opposite is not generally true.

Returning to our example it easy to see, that p_1 is neither, p_2 is closed under taking minors (see also Prop. 3.1.3), while p_3 is closed under contractions, but not under taking minors, since deleting an edge can prevent a vertex from having an adjacent vertex in the selected set.

Now let us consider an arbitrary, large enough planar graph G. We want to examine the relation between the value of parameters p_1, p_2, p_3 and the surface on which the graph expands. What effect has the a differentiation of the second to the parameters value? Moreover, where do we have to look to be convinced about the value of one parameter?

One can observe, that the behavior of the parameter p_1 is totally depending in local characteristics of the graph. The overall size of the graph is completely irrelevant, as arbitrary large graphs can have smaller maximum degree than a small graph. In addition, for a certificate of the value of p_1 it suffices to take a look at the neighborhood of the vertex with maximum degree.

Now let us continue with the second parameter, the treewidth of the graph. In general, its value highly depends on the structure of the graph; however by constant structure it does have the tendency to grow as the graph expands. And this is done in a linear manner, proportional to the diameter of the graph; by Proposition 3.1.5 as example, the treepwidth of an $m \times m$ grid is equal to m.

Finally, we discuss the third parameter, the dominating set as it is best known. As the graph expands to cover bigger part of the surface, the value of p_3 is bound to increase: since the graph is plane, not all of the new vertices can afford to be joined to vertices previously forming the dominating set. In fact, the growth of p_3 is proportional to the growth of the vertex set of the graph, as the vertices of the dominating set spread to the whole surface of the graph, following its surface.

There is a number of parameters that behave similarly, sharing interesting properties, which allow them to be approached, in terms of algorithm design, in a distinct manner. Due to their characteristic span over the surface of a graph, discussed above, they have been named *bidimensional*. More precise, we define:

Definition 4.1.1. A parameter $\mathbf{p}$ is *minor bidimensional* with density $\delta_{\mathbf{p}}$ if:

- $\mathbf{p}$ is closed under taking minors
- for the $r \times r$ grid R, $\mathbf{p}(R) = (\delta_{\mathbf{p}} r)^2 + o((\delta_{\mathbf{p}} r)^2)$.

A parameter $\mathbf{p}$ is *contraction bidimensional* with density $\delta_{\mathbf{p}}$ if:

- $\mathbf{p}$ is closed under contractions
- for any partially triangulated the $r \times r$ grid R, $\mathbf{p}(R) = (\delta r)^2 + o[(\delta r)^2]$
- δ_P is the smallest δ among all partially triangulated $r \times r$ grids.

In either case, $\mathbf{p}$ is called *bidimensional*. The *density* $\delta_{\mathbf{p}}$ of $\mathbf{p}$ is the minimum of the two possible densities (when both definitions are applicable) and generally holds $0 < \delta_{\mathbf{p}} \leq 1$.

4.2 Bidimensional Parameters

Let us introduce, next, a selection of important and well known graph problems, that have rightful place amongst the numerous examples of bidimensional parameters and discuss their properties:

Vertex Cover (vc). A *Vertex Cover* of a graph is a set of vertices, such that every edge of G has at least one endpoint in the vertex cover. The *vertex cover number* of a graph G, denoted as $\mathbf{vc}(G)$, is the size of a minimum vertex cover of G. The p-Planar Vertex Cover problem is to decide, given a planar graph G and a positive integer k, whether G has a vertex cover of size at most k. Let us note that vertex cover has density $\delta_{\mathbf{vc}} = 1/\sqrt{2}$ and is closed under taking minors, i.e. if a graph G has a vertex cover of size k, then each of its minors has a vertex cover of size at most k.

Feedback Vertex Set (fvs). A *Feedback Vertex Set* of a graph G is a set of vertices, such that the subgraph induced by the vertices of G not belonging to the feedback vertex set, has no cycles. The *feedback vertex set number* of a graph G, denoted as $\mathbf{fvs}(G)$, is the minimum size of a feedback vertex set of G. The p-Planar Feedback Vertex Set problem is to decide, given a planar graph G and a positive integer k, whether G has a feedback vertex set of size at most k. Feedback vertex set has density $\delta_{\mathbf{fvs}} \in [1/2, 1/\sqrt{2}]$ and is closed under taking minors.

Face Cover (fc). A *Face Cover* of a plane graph G is a set of faces, such that all vertices of G are lying on the boundary of one of the faces in the face cover. The *face cover number* of a graph G, denoted as $\mathbf{fc}(G)$ is the size of a minimum face cover of G. The p-FACE COVER problem is to decide, given a graph G and a positive integer k, whether G has a face cover of size at most k. The parameter face cover has density $\delta_{\mathbf{fc}} = 1/2$ and is closed under taking minors.

Dominating Set (ds). A *Dominating Set* of a graph G is a set of vertices, such that every vertex outside the dominating set is adjacent to a vertex in it. The *dominating set number* of a graph G, denoted as $\mathbf{ds}(G)$, is the size of a minimum dominating set of G. The p-PLANAR DOMINATING SET problem is to decide, given a planar graph G and a positive integer k, whether G has a dominating set of size k. This parameter is previously discussed; it has density $\delta_{\mathbf{is}} = 1/3$ and is closed under edge contractions, but not under taking minors.

Longest path (lp).. A *Longest Path* of a graph G is a path of maximum length and the *longest path number*, denoted as $\mathbf{lp}(G)$, is this length. The p-LONGEST PATH problem is to decide, given a graph G and a positive integer k, whether G contains a path of length at least k. The parameter longest path is closed under taking minors and has density $\delta_{\mathbf{lp}} = 1$.

Cycle Packing (cp). The *cycle packing number* of a graph G, denoted as $\mathbf{cp}(G)$, is the maximum number of disjoint cycles in G. The p-Planar Cycle Packing problem is to decide, given a planar graph G and a positive integer k, wether $\mathbf{cp}(G) \leq k$.

4.3 The Win/Win Approach

The standard technique for the design of subexponential parameterized algorithms for graph parameters on planar graphs, relies on two conditions: the existence of a sublinear combinatorial bound for the branchwidth in terms of the parameter and dynamic programming on branch decompositions.

Roughly speaking, the idea of the Win/Win approach is the following: Goal is an algorithm to answer the question, wether the value of a parameter for a planar graph is greater or less than a given integer. We will try to acquire a function of the parameter which upper bounds the branchwidth, and an algorithm that can solve the problem for "small" branchwidth. Then, we compute the branchwidth of the given graph (recall that this is tractable, since the graph is planar); if the branchwidth is very large, we can immediately deduce, that the value of the parameter is also accordingly big, else we have small branchwidth and we can proceed in calculating the value of the parameter (see also [10]).

To make this precise, we refer to any graph parameter $\mathbf{p}$, for which there exist two positive real numbers α and β, such that:

(A) For any planar graph G, $\mathbf{bw}(G) \leq \alpha \cdot \sqrt{\mathbf{p}(G)} + O(1)$.

(B) For every graph planar G and given an optimal branch decomposition of G, $\mathbf{p}(G)$ can be computed in $O(2^{\beta \cdot \mathbf{bw}(G)} \cdot n)$ steps.

The following theorem ensures the existence of an algorithm deriving from these two conditions, to which we will refer for the rest of the study without further notice as Condition (A) and Condition (B), formalizing thus the discussion preceded.

Theorem 4.3.1. *Let* $\mathbf{p}$ *be a parameter and suppose that Conditions (A) and (B) are satisfied for some constants* α *and* β*, respectively. Given an input* (G, k) *and an optimal branch decomposition of* G*, one can solve the associated with* $\mathbf{p}$ *parameterized problem in* $O(2^{\alpha \cdot \beta \cdot \sqrt{k}} n)$ *steps.*

Proof. Given the optimal branch decomposition of G, we first check whether $\mathbf{bw}(G) > \alpha \cdot \sqrt{k}$. If this is true, then by Condition (A) we have equivalently $\alpha \cdot \sqrt{\mathbf{p}(G)} > \alpha \cdot \sqrt{k}$, i.e. $\mathbf{p}(G) > k$. Hence, the answer to the according minimization problem is "yes"(to the maximization problem "no"). Else, by Condition (B), using the given branch decomposition we can calculate $\mathbf{p}(G)$ in $O(2^{\alpha \cdot \beta \cdot \sqrt{k}} n)$ steps. □

Recall that an optimal branch decomposition of any planar graph can be computed in $O(n^3)$ steps (see also Paragraph 3.2). Therefore, whenever we apply Theorem 4.3.1 without assuming the existence of an optimal branch decomposition, we should add an additive overhead of $O(n^3)$ steps to the complexity of the algorithm.

Thus, the existence of an efficient algorithm for a parameterized problem associated to a parameter $\mathbf{p}$, depends on wether the fulfillment of Conditions (A) and (B) can be guaranteed, for some constants α and β, respectively. If that is the case for a parameter $\mathbf{p}$, we denote as $\alpha_{\mathbf{p}}$, $\beta_{\mathbf{p}}$ the minimum values of α, β for which the two conditions hold. The objective of the next two paragraphs is to present how Conditions (A) and (B) can be satisfied, and how the values of the constants can be determined.

4.4 Dynamic Programming

One established technique for attacking NP-hard problems is dynamic programming. We already saw in Chapter 3, that graph decompositions may come useful when attempting to apply dynamic programming. The separation properties of the decompositions allow addressing a subdivision of the problem in question. As known, dynamic programming takes advantage of recurrent structure in overlapping subproblems to optimize the running time of the algorithm.

Similar procedures can be implemented often to different kind of graph decompositions ([4, 17, 19]). Here, we are mainly interested in using branch decompositions.

We present, as example, a sketch of an algorithm computing the vertex cover of a planar graph of bounded branchwidth. We assume that an optimal branch decomposition (T', μ) of the input graph G of m edges is given. (Recall that, otherwise, one can be constructed in $O(n^3)$ steps [46, 40]). Let $\omega' : E(T') \rightarrow 2^{V(\mathcal{G})}$ be the order function of (T', μ).

We choose arbitrarily an edge $\{x, y\}$ in T', and subdivide it by inserting a new vertex q on this edge; we add a new vertex r and an edge $\tilde{e}$ joining r and q. We denote the resulting tree as T, and by choosing r as a root, T is now considered rooted. The order function ω associated with the tree T, is defined as follows: $\omega(\{x, v\}) = \omega(\{v, y\}) = \omega'(\{x, y\})$, $\omega(\{r, v\}) = \emptyset$ and $\omega(e) = \omega'(e)$ for every other edge of T. The pair (T, μ) remains a branch decomposition of G of minimum width.

The root r imposes an ordering of the edges of T, according to which each edge $e \in T$ has two "descendent" edges; we denote them as e_1 and e_2. The subset of the edge set $E(T)$ containing e and all its descendent edges, induces a connected subtree of T, which we denote as T_e. The leaves of the subtree T_e are mapped by μ^{-1} to edges of the graph, which induce a subgraph of G, denoted as G_e. Observe, that $T_e = T_{e_1} \cup T_{e_2} \cup e$ and $G_e = G_{e_1} \cup G_{e_2}$.

For each edge of e of T, we define the evaluation function B_e on a subset of the vertex set of the input graph G and a positive integer, as follows:

$$B_e(S,k) = \begin{cases} 1 & \exists R : R \text{ is a } VC \text{ in } G_e \ \& \ |R| = k \ \& \ R \cap \omega(e) = S \\ 0 & \text{otherwise} \end{cases}$$

Let now e be an arbitrary edge of T. The value of B_e is passed on from the values B_{e_1} and B_{e_2} of the descendent edges of e, obeying to the following rule:

$$\begin{aligned} B_e(S,k) = 1 \Leftrightarrow \exists k_1, k_2 \leq k, \exists S' \subseteq (\omega(e_1) \cup \omega(e_2) - \omega(e)) : \\ B_{e_1}(S' \cup (S \cap \omega(e_1), k_1) = 1 \ \wedge \ B_{e_2}(S' \cup (S \cap \omega(e_2), k_2) = 1 \\ \wedge \ k_1 + k_2 - |S'| - |\omega(e) \cap \omega(e_1) \cap \omega(e_2)| = k \end{aligned}$$

Therefore, to reply whether the input graph G has a vertex cover of size at most k, we calculate the value $B_{\tilde{e}}(\emptyset, k)$ (recall that $\tilde{e} = \{r, q\}$ where r the root of T).

The algorithm requires $k \cdot 2^{|\omega(e)|}$ steps for the evaluation of B_e for each of the $k \cdot 2^{|\omega(e)|}$ possible selections of vertices in $\omega(e)$ that belong to the vertex cover of G_e; and that for each edge e of T. Hence, we have an $O(m \cdot (\mathbf{bw}(G))^2 \cdot 4^{\mathbf{bw}(G)})$ step algorithm deciding the problem.

In general, the running time of an algorithm of similar structure, computing a bidimensional parameter given a planar graph G of bounded branchwidth, is of the form $O(2^{O(\mathbf{bw}(G))} \cdot n)$, meeting thus the demands of Condition (B). The constants depend on the specific parameter, but also highly on choice for the encoding of the middle sets, which may affect the processing time significantly. Algorithms based on dynamic programming have been studied extensively on a number of bidimensional parameters by Dorn in [16]. By implementation of a non-trivial technique involving fast matrix multiplication, it is concluded among other, that for any n-vertex planar graph G, given an optimal branch decomposition:
p-Planar Vertex Cover, p-Planar Dominating Set, p-Planar Feedback Vertex Set and p-Planar Cycle Packing can be solved in $O(2^{1.19 \cdot \mathbf{bw}(G)} \cdot n)$, $O(2^{1.89 \cdot \mathbf{bw}(G)} \cdot n)$, $O(2^{3.56 \cdot \mathbf{bw}(G)} \cdot n)$ and $O(2^{2.78 \cdot \mathbf{bw}(G)} \cdot n)$ steps, respectively.

It follows immediately that, Condition (B) is satisfied for the parameters of vertex cover, dominating set and feedback vertex set for $\beta_{\mathbf{vc}} \leq 1.19$, $\beta_{\mathbf{ds}} \leq 1.89$, $\beta_{\mathbf{fvs}} \leq 3.56$ and $\beta_{\mathbf{cp}} \leq 2.76$, respectively.

To estimate $\beta_{\mathbf{fc}}$, we use the well known reduction of the FACE COVER problem to the PLANAR BLUE-RED DOMINATING SET problem. The later asks, given a planar bipartite graph $H = (B \cup R, E)$ and a non-negative integer k, if there is a $D \subseteq R, |D| \leq k$, such that every vertex in B has a neighbour in D. In fact it is easy to verify (for example, see [31]) that $\mathbf{fc}(G) \leq k$, if and only if (R_G, k) is a yes-instance of the PLANAR BLUE-RED DOMINATING SET (just set $B \leftarrow V(G)$ and $R \leftarrow F(G)$).

From [16, Theorem 2.3.2], PLANAR BLUE-RED DOMINATING SET can be solved in $O(2^{1.19 \cdot \mathbf{bw}(H)} \cdot |V(H)|)$ steps, provided that an optimal branch decomposition is given. Combining this fact with Lemma 3.4.5, we have that Condition (b) holds for $\beta_{\mathbf{fc}} \leq 2.38$.

It has became clear by now, that using dynamic programming on the structure a graph decomposition provides, can deliver fast algorithms for a variety of planar graph problems, as long we can come up with satisfactory bounds for the width of the decomposition. Indeed, for most practical applications, the bounds must be significantly compact (since the dependence is exponential), placing the challenge of this approach on the chase towards tighter bounds.

For that task, we are aiming to benefit from the features of the bidimensionality theory. To this goal we are dedicating the last paragraph of this chapter analyzing a generic method and the next chapter for a more thorough examination.

4.5 Grid Theorem for Planar Graphs

As we have seen in the previous paragraph, being able to bound the branch-width in instances of the previously mentioned problems is a big step in designing an efficient algorithm. Here, we see how employing results from the Graph Minor theory of Robertson and Seymour can deliver such upper bounds.

A pivotal theorem of the Graph Minor theory, states that any graph (not necessarily planar) of large enough branch-width has an arbitrary large grid as a minor. How large would be "large enough", depends of course on the size of the wanted grid minor, but is in fact so much more, that the result is of pure theoretical significance. If we restrain in dealing with planar graphs on the other hand, the theorem becomes very interesting in terms of practical use, as well. The Grid Theorem for planar graphs, as it is known, states:

Theorem 4.5.1. *Let $\ell \geq 1$ be an integer. Every (planar) graph of branch-width more than $4\ell - 3$ has a $\ell \times \ell$ grid minor.*

But how can this contribute to finding upper bounds of the branch-width? Let us assume, that we are given a graph, together with the value of one bidimensional parameter of the graph. Now, we have seen that the certificate of the solution to all problems related to bidimensional parameters, is spread over the grid. The idea is to examine in which way this is done and by doing so determine the density δ. Then we can calculate the relative size of a grid that would be too big to fit as a minor in our graph and thus, by applying the grid theorem, forcing the graph to have bounded branch-width.

To make this clear, let us demonstrate this method to produce an upper bound for a graph, in relation to the size of its dominating set, as example of a bidimensional parameter. We will show:

Proposition 4.5.2. *Let G be a (planar) graph with a dominating set D of size at most k. Then $bw(G) \leq 12\sqrt{k} + 9$.*

Proof. First remember that the dominating set parameter is not closed under taking minors, but only closed under contraction of edges. So, to the biggest $\rho \times \rho$ grid that can be a minor of our graph, we need to add all the edges of G that have been deleted; we call this (partially triangulated grid) graph H. By a sequence of decontracting edges we get the original G, so we have that $H \preceq_c G$ and thus H has also a dominating set D' of size at most k.

So, how many vertices of H can be dominated by one vertex v in D'? Clearly, if v is a non-boundary vertex of the grid, the answer is at most 9: vertex v itself, plus the eight rest vertices of H that belong to the four cells of the grid to which also v belongs. This justifies, besides, the density of the parameter $\delta_{\mathbf{ds}} = 1/\sqrt{9} = 1/3$ (see also Paragraph 4.2). That means that $(\rho - 2)^2 \leq 9k$, i.e. $\rho \leq 3\sqrt{k} + 2$.

And since G cannot have a $(\rho + 1) \times (\rho + 1)$ grid and by applying Theorem 4.5.1 for $k = 3\sqrt{k} + 3$ we get that G has branch-width of at most $4k - 3 = 12\sqrt{k} + 9$. □

From Proposition 4.5.2 follows directly that Condition (A) is satisfied for the parameter of dominating set for $\alpha = 12$. In other words, it holds that $\alpha_{\mathbf{ds}} \leq 12$. Similar, the Grid Theorem for planar graphs (4.5.1) implies in general:

Lemma 4.5.3. *If a parameter* $\mathbf{p}$ *is bidimensional with density* δ_P*, then* P *satisfies property for* $\alpha = 4/\delta_P$*, on planar graphs.*

Recalling the density values described in Paragraph 4.2, we derive as example, that $\alpha_{\mathbf{vc}} \leq 4\sqrt{2}$, $\alpha_{\mathbf{fc}} \leq 8$ and $\alpha_{\mathbf{fvs}} \leq 8$.

Chapter 5

Improving the Upper Bounds

In the last paragraph we witnessed a generic method for producing upper bounds for the branchwidth of a graph, according to the value of one of its bidimensional parameters, based on a result of the Grid Minor theory. Its wide range of application, though, is unavoidably followed by the incompetence to take advantage of the characteristics of a specific problem. Examining the structure of a graph with a given property, and comprehending in depth its features, enables us the to determine far better bounds for most of the cases. This has been done for the cases of the parameters of the vertex set and the dominating set. The derived bounds are $\mathbf{bw}(G) \leq 3.675\sqrt{\mathbf{vc}(G)}$ ([13]) and $\mathbf{bw}(G) \leq 6.364\sqrt{\mathbf{ds}(G)}$ ([35]), respectively, where G is any planar graph. In this chapter, we focus in proving upper bounds relations for the branchwidth of planar graphs in terms of the parameters of face cover, feedback vertex set and cycle packing, which lead to the improvement of the algorithmic analysis of the corresponding problems.

5.1 A "Tailor Made" Technique

The idea is, based to the structure a graph can have, regarding a specified bidimensional parameter, to construct a far more simpler graph, one that we will be calling *reduced* graph, that encapsulates though the whole structural properties of the original graph. And exactly because of this fact, the branchwidth of the reduced graph will only differ by a small factor from the branchwidth of the original graph.

To get more precise, given a planar graph G and considering a bidimensional parameter, we know by the very same essence of bidimensionality, that the certificate $S \subseteq V(G)$ of the value of this parameter is scattered over the whole graph. If we study the meaning of parameter carefully, we start to understand the characteristics of the structure of the graph around the vertices in S.

We then can hope to construct a graph mostly (if not only) on the vertices of S, we call *reduced* graph of G and denote as $red(G)$, which continues to carry all the information about the structure of G. If we succeed this, then the branchwidth, which also almost entirely depends on the structure rather than the size of a graph, is bound to be similar for both G and $red(G)$.

But the task of founding an upper bound for the reduced graph is not difficult, regarding its small size. It will, actually, suffice to bound it by a function on the size of its vertex set, namely $|S|$. In deed, we can easily attain a function of sublinear dependence to the vertex of a planar graph bounding its branchwidth.

In particular, an upper bound can be delivered directly from the grid theorem for planar graphs (Theorem 4.5.1): any planar graph G is either a grid itself, or contains a grid of size at most $\lfloor\sqrt{|V(G)|}\rfloor$; either way it holds $bw(G) \leq 4\sqrt{|V(G)|} - 3$. However, this inequality can be improved. According to the tightest bound known so far ([36]):

Theorem 5.1.1. *For any planar graph G,* $\mathbf{bw}(G) \leq \sqrt{4.5} \cdot \sqrt{|V(G)|}$.

This non-trivial proof is based on a relation between slopes and majorities, the two notions introduced by Robertson & Seymour in [44] and Alon, Seymour & Thomas in [2], respectively.

5.2 Face Cover

Following the technique discussed above, we are now going to closely examine the parameter Face Cover. After studying the structure that this parameter enforces to a graph, we will take advantage of notions described in previous chapters, to prove one of the main results of this work, namely that the branch-width of any (planar) graph with a face cover of size at most k, is at most $2 \cdot \sqrt{4.5k}$.

Let us recall, that a face cover of a plane graph G is a set $S_G \subseteq F(G)$ of faces, such that all vertices of G are lying on the boundary of some face in S_G and that the value of the respective parameter, $\mathbf{fc}(G)$, is equal to the minimum size of a face cover of G (see also paragraph 4.2).

Given a plane graph G and a face cover S_G of it, we will refer to the faces in $S_G \subseteq F(G)$, as *$\mathcal{FC}$-faces*. We say that two $\mathcal{FC}$-faces f_1 and f_2 *touch*, if $\widehat{f_1} \cap \widehat{f_2} = \emptyset$. Two vertices will be called a *pair*, if they are adjacent and lie on the same $\mathcal{FC}$-face. We call a face of G *triangle*, if it has exactly three vertices on its boundary. We call an edge e in G *bridge*, if there are two $\mathcal{FC}$-faces f_1 and f_2, such that e is the unique edge, having one endpoint in the boundary of f_1 and the other in the boundary of f_2.

Let f_1, f_2 be two $\mathcal{FC}$-faces and let x_1, x_2, y_1, y_2 be four vertices such that $x_i, y_i \in \widehat{f_i}$, for $i = 1, 2$; a noose of the form $x_1 y_1 x_2 y_2 x_1$, will be called a *4-noose*. As a Jordan curve, a 4-noose N bounds two open discs. If one of them contains exactly one hyperedge, whose endpoints are the four vertices on N, then we refer to such a 4-noose as *trivial*.

Normalization. The first task is to normalize the input graph; that is, after applying a series of transformations, to bring it into a stable form, which exposes the structure of the graph and at the same time, enables us to process it in the steps to follow. This involves adding of edges and vertices, so that at the end, the input graph is a minor of the resulting one.

Let us, first, define an operation we will to need, where a vertex is replaced by two new vertices and its incident edges are distributed among them two. So, let G be a plane graph, and $x \in V(G)$ a given vertex, with at least four incident edges. We denote as $e_1, e_2, \ldots, e_n$ the edges incident to x, listed according their cyclic order in the drawing of G. For some indices $\ell < m < n$, we say that $e_\ell, e_{\ell+1}$ and e_m, e_{m+1} *split x into* x_1, x_2, to denote the following operation:

We label as v_i the other (besides x) endpoint of e_i in G, for $i = 1, \ldots, n$. We fix an open disc Δ_x containing x, whose boundary intersects all incident edges to x and nothing else. Then, remove x (and all its incident edges), and add two new vertices x_1 and x_2 in Δ_x. Finally, we add without creating crossings (note that this is doable) a new edge joining x_1 and x_2, as well as following edges:

$$e'_i = \begin{cases} x_1 v_i & , 1 \leq i \leq \ell \\ x_2 v_i & , \ell < i \leq m \\ x_1 v_i & , m < i \leq n \end{cases}$$

We can, now, proceed to the first lemma, which assures us, that we can built up the input graph into another, carrying the desired structural properties:

Lemma 5.2.1. *Let G be a 3-connected simple plane graph such that $\mathbf{fc}(G) \leq k$. Then, there exists a plane graph G' and a face cover $S_{G'}$ of G', such that:*

(a) $G \leq G'$.

(b) $|S_{G'}| \leq k$.

(c) *G' is simple and 3-connected.*

(d) *No two different $\mathcal{FC}$-faces touch.*

(e) *G' does not contain any bridge.*

(f) *A face of G' is either a $\mathcal{FC}$-face, or a square whose boundary contains two pairs of two different $\mathcal{FC}$-faces or a triangle incident to three different vertices that in turn are incident to three different $\mathcal{FC}$-faces.*

Proof. Let S_G be a face cover of G, with $|S_G| \leq k$. Note that $\mathbf{fc}(G)$ is at least 2, as otherwise G would not be 3-connected, and that all faces have at least three vertices on their boundary, due to the assumption of no multiple edges. We will consecutively apply a number of transformations forming the graph G', that has the desired properties. At each step, our purpose is that the resulting graph can inherit the face cover of the preceded graph.

• **Face detachment.** *i) Until no two $\mathcal{FC}$-faces share an edge: let e be an edge, such that $\widehat{f_1} \cap \widehat{f_2} = e$ for two $\mathcal{FC}$-faces f_1, f_2, and that an endpoint x of e lies on no $\mathcal{FC}$-face other than these two; then duplicate e into e_1, e_2, and let e_1, e_2 and $\tilde{e}'_1, e'_1$ split x into x_1, x_2 (where e'_1 is the other edge on the boundary of f_1 incident to x and $\tilde{e}'_1$ the next incident edge according to the cyclic order).*
ii) While two $\mathcal{FC}$-faces f_1, f_2 share a vertex y: label as e, e' the two edges on the boundary of f_1 incident to y and as $\tilde{e}, \tilde{e}'$ the next, incident to y, edges of e, e' respectively (according to the cyclic order); let $e, \tilde{e}$ and $e', \tilde{e}'$ split x into y_1, y_2.

First of all, notice that if two $\mathcal{FC}$-faces f_1 and f_2 of G touch, then the set $\widehat{f_1} \cap \widehat{f_2}$ is either an edge or a vertex of G, as otherwise, G cannot be simple and 3-connected, and that in the first case there always exist such an edge, that enables us to apply the transformation.

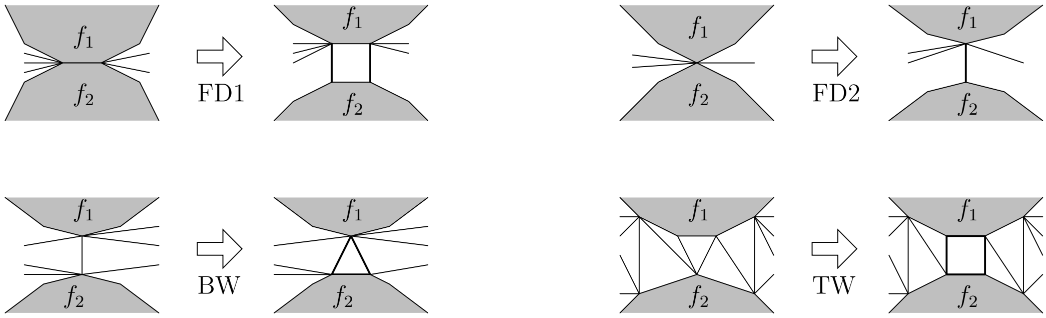

Figure 5.1: The transformations of the proof of Lemma 5.2.1.

Consider, now, somewhere in this process, a vertex v on the boundary of the $\mathcal{FC}$-face f_1, f_2 is about to be split into v_1, v_2. By the definition of this operation, after the splitting, a face $f_1^{\#}$ will have on its boundary the same vertices as f_1 with the exemption that v is replaced by either v_1 (similar for $f_2^{\#}$). Then, let the existing face cover of the graph before the split, be passed on to the resulting graph, where f_1 is replaced by its corresponding face $f_1^{\#}$, and f_2 by $f_2^{\#}$. This forms a chain, so that the resulting graph at the end of the process, G_1, has a face cover S_1 inherited in this way by S_G; there is therefore a bijection $S_G \to S_1$ and hence, (b) holds for the graph G_1.

Furthermore, each repetition of this transformation, as the opposite of an contraction, could not harm the 3-connectivity anywhere else, but among the new vertices produced by splittings of a vertex; since these vertices still lie on a cycle - the boundary of an $\mathcal{FC}$-face and are joined to neighbors of the old vertex (which were surely pairwise linked by paths disjoint to this $\mathcal{FC}$-face, because the preceded graph was 3-connected), there is no set of two vertices separating the graph. In addition the graph remains after each repetition simple, due to the splitting of the vertex in question. Thus, G_1 satisfies (c).

The transformation ends, when the boundaries of the $\mathcal{FC}$-faces in G_1 are pairwise distinct, so we can conclude that (d) is also true. Note, that G is a minor of G_1.

• **Partial triangulation.** *We add edges in the faces that don't belong in the face cover, without creating crossings, until all faces are either triangles or $\mathcal{FC}$-faces.*

Let G_2 be the resulting graph. Since all we did was adding edges, G_1 is a minor of G_2, the 3-connectivity cannot be harmed and $S_2 = S_1$ is a face cover of G_2, as well. All other faces are triangles with their boundary vertices either on two or on three $\mathcal{FC}$-faces (a triangle with all its boundary vertices on the boundary of one $\mathcal{FC}$-face, would imply that two of them separate the third one). Now, if G_2 has a double edge, then it forms a closed line that bounds two open discs; none of them is a face, so at least one vertex lies in each, which contradicts that G_1 is 3-connected. So (b), (c) and (d) are satisfied for the graph G_2.

• **Bridge widening.** *While there exist a bridge e: label as x one endpoint of e, as f the $\mathcal{FC}$-face, for which $x \in \widehat{f}$ and as e_1, e_2 the two edges in $\widehat{f}$ incident to x; duplicate by e adding the edge e', and let e, e' and e_1, e_2 split x into x_1, x_2.*

Let G_3 be the resulting graph. Similar as in the first transformation, G_3 inherits a face cover S_3 of size at most k, is simple and 3-connected, and has still no touching $\mathcal{FC}$-faces. Hence, conditions (b), (c) and (d) are true for G_3. By the end of this transformation, G_3 satisfies also (e). In addition the graph G_3 is a minor of G_3.

In the place of each bridge we create a triangle face, since the one endpoint of the bridge is split into two vertices, both joined to the vertex, that was the other endpoint of the bridge before the operation. Furthermore, the two faces on the boundary of whose the bridge lay, where triangles; after the transformation they still are. Finally, all other faces not in the face cover remain unaffected. These facts imply, that every face of the graph G_3 is either a $\mathcal{FC}$-face, or a triangle with its boundary vertices on two or three $\mathcal{FC}$-faces.

• **Triangle widening.** *While there exists a triangle face with boundary vertices x_1, x_2, y lying on exactly two disks (x_1, x_2 being a pair): label as e_1, e_2 the edges, incident to y, on the boundary of the $\mathcal{FC}$-face that contains y; then, let x_1y, x_2y and e_1, e_2 split y into y_1, y_2.*

Let G' be the resulting graph. The graph G_3 is a minor of G' and by the transitivity of the minor relation G is a minor of G', yielding that (a) is true. Similar as before, G' inherits a face cover $S_{G'}$ of size at most k, remains simple and 3-connected, without touching faces or bridges, satisfying thus (b), (c), (d) and (e). Faces that were triangles with their boundary vertices on two $\mathcal{FC}$-faces, are transformed into squares, whose boundary contains two pairs on the boundary of these two $\mathcal{FC}$-faces. The argumentation analyzed at the end of the last transform, implies that all faces that were triangles with their boundary vertices on different $\mathcal{FC}$-faces, remain as such. We conclude, that G' satifies (f), which completes the proof. □

Our next lemma clarifies the situation, delivering a hypergraph containing no more trivial edges. In fact, all hyperedges are of a distinctive type, allowing us to comprehend in bigger depth the structure of the input graph:

Lemma 5.2.2. *Let G be a 3-connected simple plane graph such that $\mathbf{fc}(G) \leq k$. Then, there exists a plane hypergraph H and a face cover S_H of H, such that:*

(i) $\mathbf{bw}(G) \leq \mathbf{bw}(H)$.

(ii) $|S_H| \leq k$.

(iii) No two different $\mathcal{FC}$-faces touch.

(iv) H contains no edges, and each hyperedge of H has arity four, containing two disjoint pairs that are incident to two different $\mathcal{FC}$-faces.

(v) A face of H is either a $\mathcal{FC}$-face, or a degenerate one, or a triangle incident to three different vertices that in turn are incident to three different $\mathcal{FC}$-faces.

Proof. Since the given plane graph G is simple and 3-connected, the requirements of Lemma 5.2.1 are satisfied; by applying it, we obtain a plane graph G' carrying the properties (a)–(f). We will construct a hypergraph H on the vertices of G', satisfying conditions (i)–(v):

> Let f be a square face of G'. By the last property, its boundary contains four vertices, namely two pairs on two $\mathcal{FC}$-faces, say x_1, y_1 and x_2, y_2. We add a hyperedge e, so that $e \subseteq f$ and $bor^*(e) = \{x_1, x_2, y_1, y_2\}$. We carry out this procedure for every square face f in G' and complete the construction, by removing all edges.

Let us, then, check that H is as desired. First of all, note that by construction, condition (iv) is satisfied. Next, recall that, since every hyperedge of H is drawn in a different face of G', and takes the place of the edges on the boundary of the corresponding face of G', the hypergraph H is generated by G', and thus plane with branchwidth not less than this of G' (see also Paragraph 3.2). Combined with property (a), this yields that (i) is true.

Consider, now, an $\mathcal{FC}$-face f in G': By (f), each edge on the boundary of f (joining a pair), lay also in the boundary of a square face. By construction of H, the pair joined by this edge, is among the endvertices of a hyperedge h in H, and thus is joined in H by an arc $\gamma \subseteq bor(h)$. This applies to any edge of G' on the boundary of f, implying a correspondence between an edge $e \subseteq \widehat{f}$ and an arc $\gamma_e \subset bor(h_e)$. In this way, the vertices and edges in the boundary of f correspond to a boundary of a face $f_H$in H. We fix this bijection σ_{FC} between a face f in $S_{G'}$ and a face f_H in H. We can now choose:

$$S_H = \{f_H \in F(H) : \exists f \in S_{G'} \subseteq F(G') : f_H = \sigma_{FC}(f)\}$$

As every vertex in G' lay in the boundary of an $\mathcal{FC}$-face, all vertices in H lie on the boundary of a face in S_H, the last set is indeed a face cover of H. Moreover, its size is equal to the size of $S_{G'}$, and hence at most k as condition (ii) demands. In addition, bijection σ_{FC} implies that any two $\mathcal{FC}$-faces in H have no vertex in common, and (iii) is also true.

Now consider a triangle face in G'. Each of the three edges on its boundary, belongs also to the boundary of a square face, as by (f) the boundary of the triangle contains three vertices on the boundary of three different $\mathcal{FC}$-faces, and by (e) the graph G' has no bridges. Continuing as before, we can fix a bijection σ_{TR} between a triangle face in G' and the corresponding face in H, denoted again as *triangle*.

These two bijections σ_{FC} and σ_{TR}, combined with the fact, that all square faces of G' have been, virtually, replaced by hyperedges in H, implicates that a face in H is either a $\mathcal{FC}$-face, or a face bounded by two hyperedges (a degenerate one), or a triangle incident to three vertices, each on a different $\mathcal{FC}$-face; thus condition (v) is satisfied, which completes the proof. □

A plane hypergraph H with a face cover S_H, satisfying properties (iii)–(v) of Lemma 5.2.2, will be characterized, from now on, as *normalized*.

Lemma 5.2.3. *Let H be a normalized hypergraph with face cover S_H and let N be a non-trivial 4-noose bounding the closed discs Δ_1, Δ_2. Let also H_i, ($i = 1, 2$) be the subgraph of H containing all vertices and edges included in Δ_i, plus the edge $\tilde{e}$ with endpoints the four vertices the 4-noose passes through. Then H_i (for $i = 1, 2$) is a normalized graph with $\mathbf{fc}(H_i) \leq \mathbf{fc}(H)$ and less vertices than H.*

Proof. Let us label the noose N as $x_1y_1y_2x_2x_1$, where $x_j, y_j \in d_j$ (for $j = 1, 2$) and d_1, d_2 two $\mathcal{FC}$-faces. Note that none of the x_j, y_j can be a pair, as otherwise they would both be pairs and all four vertices would lie on a hyperedge, contradicting that N is non-trivial. Let w_j, z_j be two vertices of f_j, that keep x_j and y_j from being a pair; they lie, then, in different open discs bounded by N, which implies that G_i (for $i = 1, 2$) has less vertices than G, as wanted.

Notice that f_j ($j = 1, 2$) is divided by N into two faces $f_j^i := f_j \cap \Delta_i$ for $i = 1, 2,\ j = 1, 2$. The faces f_1, f_2 are the only $\mathcal{FC}$-faces crossed by N and hence, we can choose the face cover S_i of H_i ($i = 1, 2$) as follows:

$$S_i = \{f_1^i, f_2^i\} \cup \{f \in S_H : f \subseteq \Delta_i\}, \quad i = 1, 2.$$

This guaranties, that $\mathbf{fc}(H_i) \leq \mathbf{fc}(H)$ for $i = 1, 2$. It remains now to verify that conditions (*iii*)–(v) of Lemma 5.2.2 stay invariant in both H_1, H_2. First, since all faces of H_i, ($i = 1, 2$) are subsets of the corresponding faces in H, we can deduce that (*iii*) is true.

Observe that no edges are added to H_1, H_2. All hyperedges of the two graphs except $\tilde{e}$, already existed in H and so, they comply with the demands of (*iv*); the newly added hyperedge $\tilde{e}$ contains the pairs x_1, y_1 and x_2, y_2, which are in turn incident to the new faces f_1^i, f_2^i (for $i = 1, 2$). Combining these facts, we conclude that condition (*iv*) holds.

All faces of H, not in the face cover, where either degenerate or triangles; thus, a part of a noose outside an $\mathcal{FC}$-face, can cross either a degenerate face creating two degenerate faces, one in each subgraph H_i, ($i = 1, 2$), or a triangle, creating a triangle in the one subgraph and a degenerate face in the other. In any case, a face of H_i, ($i = 1, 2$) that is not in the corresponding face cover, can be either a degenerate one or a triangle with boundary vertices on the boundary of three different $\mathcal{FC}$-faces, as desired; i.e. (v) is satisfied and the proof completed. □

Prime Hypergraphs. A normalized hypergraph H will be denoted as *prime*, if every 4-noose is trivial. Let H be a prime hypergraph and S_H a face cover of H with $|S_H| \geq 3$. We define its *reduced* graph $\mathbf{red}(H)$ as follows: There is a bijection $\phi_v : S_G \to V(\mathbf{red}(H))$ and a bijection $\phi_e : E(H) \to E(\mathbf{red}(H))$, such that two vertices in $x, y \in V(\mathbf{red}(H))$ are joined by an edge in $E(\mathbf{red}(H))$, if and only if there is a hyperedge with vertices lying on the faces $\phi_v^{-1}(x)$ and $\phi_v^{-1}(y)$.

Lemma 5.2.4. *Let H be a prime hypergraph with $\mathbf{fc}(H) \geq 3$. Then, the graphs H^* and $\tilde{R}_{\mathbf{red}(H)}$ are isomorphic.*

Proof. Notice that in a prime hypergraph all faces are either triangles or $\mathcal{FC}$-faces. This implies that the vertices of H^* can be partitioned to those, that correspond to $\mathcal{FC}$-faces and those that correspond to trinagles of H. We denote these two vertex sets of H^* as $V_{\mathcal{FC}}(H^*)$ and $V_{\mathcal{TR}}(H^*)$. On the other hand, the $\mathcal{FC}$-faces of H correspond to vertices of $\mathbf{red}(H)$ and the triangles of H correspond to the faces of $\mathbf{red}(H)$. Moreover the sets $V(\mathbf{red}(H))$ and $F(\mathbf{red}(H))$ correspond to the two parts of the vertex set of $R_{\mathbf{red}(H)}$ and thus to a bipartition $V_1(\tilde{R}_{\mathbf{red}(H)}), V_2(\tilde{R}_{\mathbf{red}(H)})$ of the vertices of $\tilde{R}_{\mathbf{red}(H)}$. We now have a chain of bijections that merge into a bijection σ between $V_{\mathcal{FC}}(H^*) \cup V_{\mathcal{TR}}(H^*)$ and $V_1(\tilde{R}_{\mathbf{red}(H)}) \cup V_2(\tilde{R}_{\mathbf{red}(H)})$. We claim that σ is a isomorphism from H^* to $\tilde{R}_{\mathbf{red}(H)}$. To see this, observe that any hyperedge e of H^* has four endpoints containing two anti-diametrical pairs: two corresponding to $\mathcal{FC}$-faces and two corresponding to triangles of H. Notice that these $\mathcal{FC}$-faces and triangles of H correspond to vertices and faces of $\mathbf{red}(G)$ and therefore to the vertices of the hyperedge $\sigma(e)$ of $\tilde{R}_{\mathbf{red}(H)}$. □

To gain an insight into the course of the proof, let us take a step back and reflect on the work done so far. Given a plane graph of a known face cover size, we formed a chain of transitions, relating the input graph gradually to other, of more evident structure. Stripping systematically the graph down to the bone, we progressed to a concrete version of the graph, that generated the hypergraph we denoted normalized, which in turn led us to the prime hypergraph. Through the portal of duality, and then by employing a powerful tool, namely the radial graph, we succeeded in relating the prime hypergraph to a very simple triangulated plain graph, we referred to as reduced.

We now, find ourselves at the doorway of the labyrinth's center. The reduced graph, carrying the whole structure of the input graph, and relieved of excess or unwanted components, enables us to estimate an upper bound of its branchwidth. This is accomplished in the following Corollary. Afterwards, all we have to do is pick up the Ariadne's thread, that we carefully laid behind us, so we can pass on this bound back to our input graph; this is completed by the next two lemmas, winding up the proof.

Corollary 5.2.5. *If H is a prime hypergraph, then $\mathbf{bw}(H) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(H)}$.*

Proof. If $\mathbf{fc}(H) = 2$, then H is the graph of 6 vertices - three on each disk - with 3 edges of arity of four between these vertices. It is $\mathbf{bw}(H) = 4 \leq$

$2 \cdot \sqrt{4.5 \cdot 2}$. Suppose now that S_H is a face cover of H where $3 \leq |S_H| = \mathbf{fc}(H)$ and notice that $\mathbf{red}(H)$ contains $|S_H|$ vertices. From the main result in [36], any n-vertex plane graph has branchwidth bounded by $\sqrt{4.5 \cdot n}$. Applying this result on $\mathbf{red}(H)$ we have that $\mathbf{bw}(\mathbf{red}(H)) \leq \sqrt{4.5 \cdot \mathbf{fc}(H)}$. Recall that the hypergraph H is generated by a 3-connected graph, namely the normalized graph; thus by Corollary 3.4.3 it follows that $\mathbf{bw}(H) = \mathbf{bw}(H^*)$. Combining this with Lemmata 3.4.5 and 5.2.4, we obtain that :

$$\mathbf{bw}(H) = \mathbf{bw}(H^*) = \mathbf{bw}(\tilde{R}_{\mathbf{red}(H)}) \leq 2 \cdot \mathbf{bw}(\mathbf{red}(H)) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(H)}$$

. and the desired inequality follows directly. □

Lemma 5.2.6. *Let H be a normalized graph. Then* $\mathbf{bw}(H) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(H)}$.

Proof. We use induction on the number of vertices of H. In case $|V(H)| = 6$, G has two $\mathcal{FC}$-faces, three vertices on each of them, and three hyperedges. So, indeed $\mathbf{bw}(H) \leq 4 \leq 2 \cdot \sqrt{4.5 \cdot 2}$.

We now assume, that for any normalized hypergraph H, where $6 \leq |V(H)| < n$, it holds that $\mathbf{bw}(H) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(H)}$ and we will show that the same upper bound holds for any normalized hypergraph H with n vertices.

If H is prime, then the result follows directly from Corollary 5.2.5. Suppose, then that H is not prime, and therefore contains a non-trivial 4-noose N. As N bounds two discs Δ_1, Δ_2, Lemma 5.2.3 implies that for $i = 1, 2$ the graph H_i is a normalized graph with $\mathbf{fc}(H_i) \leq k$ and $|H_i| < n$. By the induction hypothesis, we obtain $\mathbf{bw}(H_i) \leq 2 \cdot \sqrt{4.5 \cdot k_i}$, $(i = 1, 2)$. Finally, using Lemma 3.2.5, we conclude that $\mathbf{bw}(H) = \max\{\mathbf{bw}(H_1), \mathbf{bw}(H_2)\}$, i.e. $\mathbf{bw}(H) \leq 2 \cdot \sqrt{4.5 \cdot k}$. □

Theorem 5.2.7. *For any planar graph G,* $\mathbf{bw}(G) \leq 2 \cdot \sqrt{4.5} \cdot \sqrt{\mathbf{fc}(G)}$.

Proof. We can assume that $\mathbf{fc}(G) \geq 2$, as otherwise G is either a forest or an outerplanar graph, implicating that $\mathbf{bw}(G) \leq 2$ yielding trivially the result. Also, we can assume that G is simple as the removal of loops or multiples edges may reduce the branchwidth of a graph by at most 2 and this only in the case where the resulting graph is a forest.

We will use induction on $|V(G)|$. For the smallest graph with $\mathbf{fc}(G)$ at least two, namely the K_4, the upper bound is true. We assume the same for any graph with less than $n > 4$ vertices and we will show that it holds also for any n-vertex graph.

If the graph G is 3-connected, then by Lemmata 5.2.1 and 5.2.2, there is a hypergraph H, with $\mathbf{fc}(H) \leq \mathbf{fc}(G)$ and $\mathbf{bw}(G) \leq \mathbf{bw}(H)$, and the result follows directly from Lemma 5.2.6.

So, let us assume that G is not 3-connected. Then, it has a separator of at most two vertices. We have the following cases:

- The minimum separator has two vertices.

Let C be some of the connected components of $G[V(G) - \{x, y\}]$. We set $G_1 = G[V(C) \cup \{x, y\}]$ and $G_2 = G[V(G) - V(C)]$ and we add in both G_1 and G_2 the edge $e = \{x, y\}$ (if its does not already exists). Notice that $G_i \leq G$ and therefore $\mathbf{fc}(G_i) \leq \mathbf{fc}(G)$. From the induction hypothesis $\mathbf{bw}(G_i) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(G_i)}$ and the result follows by applying 3.2.5 for G_1 and G_2.

- The minimum separator is a cut vertex.

Two non-empty subgraphs, say G_1 and G_2, are joined by this vertex. For $i = 1, 2$ we have $|V(G_i)| < N$ and hence, by the induction hypothesis $\mathbf{bw}(G_i) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(G_i)}$. In addition, $\mathbf{fc}(G_i) \leq \mathbf{fc}(G)$ for $i = 1, 2$, since $G_i \leq G$. Applying Lemma 3.2.5 for the graph $G = G_1 \cup G_2$ we obtain $\mathbf{bw}(G) = \max\{\mathbf{bw}(G_1), \mathbf{bw}(G_2)\}$. Combining these facts, we have $\mathbf{bw}(G) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fc}(G)}$ and again the desired relation holds.

- The graph is disconnected.

Again, we have two subgraphs G_1 and G_2 and the proof continues similar to the previous case, completing the induction. □

5.3 Feedback Vertex Set

We turn now on one of the most famous graph problems, the Feedback Vertex Set (see e.g. [32]). Flow diagrams are being used in different fields of science, like economics, since long time. It soon arose the question, if someone could avoid circling in the diagrams, blocking thus unwanted feedback. Turning to mathematics to provide an answer, it became apparent that the problem was not simple. On the contrary, it was proved to be NP-hard, being as a matter of fact one of the first problems to be characterized so.

Recall that a *Feedback Vertex Set* of a plane graph G is a set of vertices $S \subseteq V$, such that the induced subgraph $G[V \setminus S]$ has no cycles. The feedback vertex set number of a graph G, denoted as $\mathbf{fvs}(G)$, is defined as the minimum size of a feedback vertex set in G (see also Paragraph 4.2).

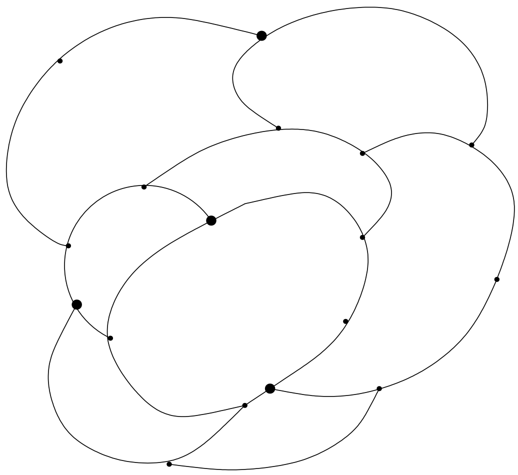

Figure 5.2: A graph with a Feedback Vertex Set.

We observe that a feedback vertex set of a planar graph, is related to a face cover of the graph: the first is a set of vertices hitting each cycle and thereby, the boundary of each face, and the second a set of faces on the boundary of whose all vertices lie. In deed, face cover and planar feedback vertex set are closely related in dual graphs. Informally speaking, the "dual" version of the face cover number is upper bounded by the vertex feedback set number:

Lemma 5.3.1. *Let G and G^* be dual plane graphs that are not forests. Then,* $\mathbf{fc}(G^*) \leq \mathbf{fvs}(G)$.

Proof. Let $S \subseteq V(G)$ be a feedback vertex set in G, with $|S| \leq k$. As the boundary of any face $f \in F(G)$ contains a cycle of G, it also contains a vertex $v \in S$. This implies that any vertex $f^* \in V(G^*)$ of G^* is in the boundary of some face v^* of S^*, where $S^* \subseteq F(G^*)$ is the set of the duals of the vertices in S. Therefore S^* is a face cover of G^*. ☐

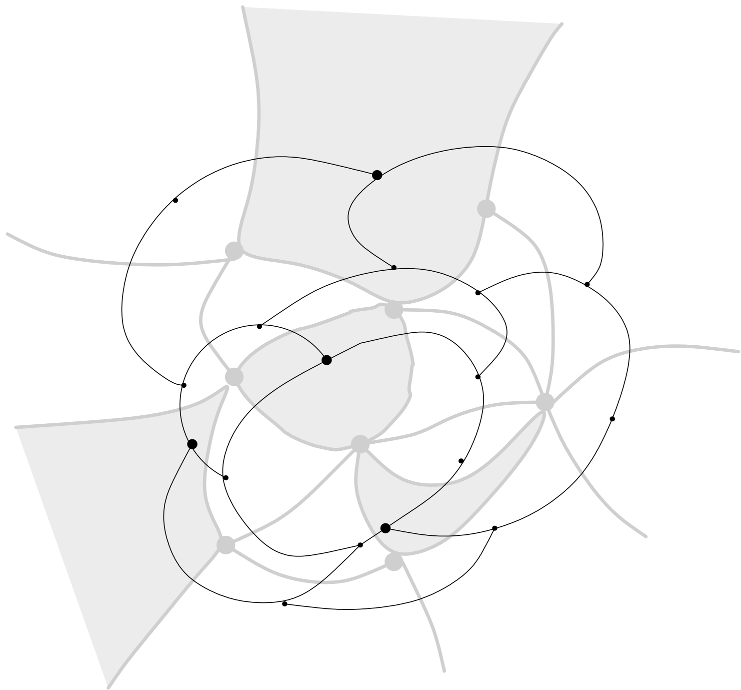

Figure 5.3: The previous graph with a Feedback Vertex Set and its dual with the corresponding Face Cover.

Therefore, the task of bounding the the branchwidth of a planar graph in relation to its feeback vertex set number can be reduced to the bound associated with the face cover number. In particular, utilizing the upper bound that we proved for the branchwidth of a planar graph G in relation to its face cover number $\mathbf{fc}$(G), we prove the following:

Theorem 5.3.2. *Let G be a plane graph with $\mathbf{fvs}(G) \geq 1$. Then its branchwidth is at most $\mathbf{bw}(G) \leq 2 \cdot \sqrt{4.5 \cdot \mathbf{fvs}(G)}$.*

Proof. Let us consider plane graphs G and its dual graph G^*. Since $\mathbf{fvs}(G) \geq 1$, none of the two graphs can be a forest. Hence, the requirement of Lemma 5.3.1 is satisfied. Combining this with Theorem 5.2.7 and the fact that dual graphs have the same branchwidth (see also Corollary 3.4.3), the following inequality is formed:

$$\mathbf{bw}(G) = \mathbf{bw}(G^*) \leq 2 \cdot \sqrt{4.5} \cdot \sqrt{\mathbf{fc}(G^*)} \leq 2 \cdot \sqrt{4.5} \cdot \sqrt{\mathbf{fvs}(G)},$$

which implies that the desired upper bound for the branchwidth of graph G holds. $\square$

Chapter 6

Algorithmic Consequences

6.1 Previous Results

We are interested in the parameterized version of the three well-known problems, namely p-Feedback Vertex Set, p-Cycle Packing and p-Face Cover, as they are defined in Paragraph 4.2.

Many FPT-algorithms were proposed for p-Feedback Vertex Set. The best current results in this direction are the $O(4^k kn)$ step probabilistic algorithm in [3] and the $O(5^k kn^2)$ step algorithm in [8].

When restricted to planar graphs, p-Planar Feedback Vertex Set, p-Face Cover and p-Planar Cycle Packing are solvable by subexponential FPT-algorithms. The first results of this kind were given by Kloks et al. in [41]. Furthermore, Fernau and Juedes proved that Face Cover can be solved in $O(2^{24.551\sqrt{k}} \cdot n)$ steps (see [31]).

6.2 Our Contribution

The theory of bidimensionality, as discussed in Chapter 4, provides the context for our analysis of the algorithms. In particular, we set in motion the Win/Win technique (see Paragraph 4.3) to derive algorithms for the p-Planar Feedback Vertex Set problem and the p-Face Cover problem. Recall that by the Win/Win approach, the existence of a fast subexponential algorithm, depends on determining tight upper bounds for the constants, so that Conditions (A) and (B) hold.

In Paragraph 4.4, we witnessed the existence of dynamic programming based algorithms on graphs of bounded branchwidth for the problems of Planar Feedback Vertex Set and Planar Blue Red Dominating Set, among others. In combination with the reduction of Face Cover to the second, we then concluded that Condition (B) is satisfied for the parameters of feedback vertex set **fvs**, and of face cover **fc**, for $\beta_{\mathbf{fvs}} \leq 3.56$ and $\beta_{\mathbf{fc}} \leq 1.19$, respectively.

By applying the Grid Theorem for planar graphs, we can relate the density of a parameter to a bound of the branchwidth of a planar graph (see also Paragraph 4.5). According to our presentation of bidimensional parameters in Paragraph 4.2, the density of the feedback vertex set parameter is $\delta_{\mathbf{fvs}} = [1/2, 1/\sqrt{2}]$ and of the face cover parameter $\delta_{\mathbf{fc}} = 1/2$. Hence, by Lemma 4.5.3, we derive that Condition (A) holds for $\alpha_{\mathbf{fvs}} \leq 8$ and $\alpha_{\mathbf{fc}} \leq 8$ for the two parameters, respectively.

Applying Theorem 4.3.1 implies the existence of an $O(2^{28.48\cdot\sqrt{k}} \cdot n + n^3)$ algorithm for the p-Planar Feedback Vertex Set problem, and the existence of an $O(2^{19.04\cdot\sqrt{k}} \cdot n + n^3)$ algorithm for the Face Cover problem (already improving the constants of previous results). Recall that the n^3 additive in the complexity of the algorithms is due to the demand of an optimal branch decomposition of the input graph.

The above estimations for $\alpha_{\mathbf{fvs}}$ and $\alpha_{\mathbf{fc}}$ can be easily further improved using known results. Kloks et al. [41] proved that for any planar graph G, there is a planar graph H containing G as a subgraph such that $\mathbf{ds}(H) \leq \mathbf{fvs}(G)$ (here by $\mathbf{ds}(H)$ we denote the minimum size of a dominating set of H). Moreover it holds that for any planar graph H, $\mathbf{bw}(H) \leq 6.364\sqrt{\mathbf{ds}(H)}$ [35]. As $\mathbf{bw}(G) \leq \mathbf{bw}(H)$, we obtain that $\mathbf{bw}(G) \leq 6.364\sqrt{\mathbf{fvs}(G)}$ and this yelds Condition (a) for $\alpha_{\mathbf{fvs}} \leq 6.364$. For $\alpha_{\mathbf{fc}}$, we need to make the following observation: Suppose that a plane graph G has a face cover $U \subseteq F(G)$ of size $\leq k$. Let H be the graph obtained from G, if for each $f \in U$ we draw a vertex v_f inside f and connect it with the vertices incident to f. Notice that the new vertices constitute a dominating set of H, of size at most k. Again, from the result of [35], we conclude that $\mathbf{bw}(G) \leq \mathbf{bw}(H) \leq 6.364 \cdot \sqrt{k}$, thus $\alpha_{\mathbf{fc}} \leq 6.364$.

According to the above, there is a $O(2^{22.66\cdot\sqrt{k}} \cdot n + n^3)$ step algorithm for the Planar Feedback Vertex Set problem and a $O(2^{15.15\cdot\sqrt{k}} \cdot n + n^3)$ step algorithm for the Face Cover problem.

Finally, the main combinatorial results of this study proved in Chapter 5, namely Theorems 5.2.7 and 5.3.2, imply that Condition (A) is satisfied for the parameters of feedback vertex set **fvs**, and of face cover **fc**, for $\alpha_{\mathbf{fvs}} \leq 4.243$ and $\alpha_{\mathbf{fc}} \leq 4.243$, respectively. Applying Theorem 4.3.1 for these values and for the mentioned values of $\beta_{\mathbf{fvs}}$ and $\beta_{\mathbf{fc}}$, yields to the fastest algorithms, as far as we know, for the two problems:

Theorem 6.2.1. *The p-*PLANAR FEEDBACK VERTEX SET *problem and the p-*FACE COVER *problem can be solved in* $O(2^{15.11\cdot\sqrt{k}} \cdot n + n^3)$ *and* $O(2^{10.1\cdot\sqrt{k}} \cdot n + n^3)$ *steps, respectively.*

We stress, that according to Bodlaender [6] (see also [7]) there exists a polynomial algorithm producing a $O(k^3)$ size kernel for the p-FEEDBACK VERTEX SET problem, when parameterized by k (i.e. an equivalent instance of the problem where the input graph has at most $O(k^3)$ vertices). Combining this fact with Theorem 6.2.1, we derive the existence of an $O(2^{15.11\cdot\sqrt{k}})+n^{O(1)}$ algorithm for p-PLANAR FEEDBACK VERTEX SET. For the p-FACE COVER, a $O(k^2)$ kernel has been reported in [41]. Therefore, p-FACE COVER can be solved in $O(2^{10.1\cdot\sqrt{k}}) + n^{O(1)}$ steps.

Another parameter, the algorithmic analysis of which our combinatorial results improve, is cycle packing. In paragraph 4.4 we witnessed that solving CYCLE PACKING in planar graphs can be done in $O(2^{2.78\cdot\mathbf{bw}(G)} \cdot n)$ steps, i.e. Condition (B) holds for $\beta_{\mathbf{cp}} \leq 2.78$. According to Kloks et al. in [41], for any planar graph G, it holds that $\mathbf{fvs}(G) \leq 5 \cdot \mathbf{cp}(G)$. Combining this with Theorem 5.3.2 yields that for any planar G, $\mathbf{bw}(G) \leq 2 \cdot \sqrt{4.5 \cdot 5 \cdot \mathbf{cp}(G)}$ and thus, Condition (A) holds for $\alpha \leq 9.49$. By Theorem 4.3.1, the p-PLANAR CYCLE PACKING can be solved in $O(2^{26.347\cdot\sqrt{k}} \cdot n + n^3)$ steps.

6.3 Open Problems

According to the Win/win approach, the algorithmic analysis of all problems examined in this paper is reduced to the problem of bounding the decomposability of a planar graph (i.e. the branchwidth) by a sublinear function of the parameter. While such general (but not optimal) upper bounds are provided by bidimensionality theory [11] better constants (and thus faster algorithms) have been achieved by a "tailor made" analysis of the parameter in the cases of *vertex cover*, *edge dominating set*, and *dominating set* (see [13, 35]). Our results for *feedback vertex set*, *face cover*, and *cycle packing* offer to the same line of research. Furthermore, specific combinatorial similarities between our proofs in Chapter 5 and the proofs in [13, 35], make us believe, that a generic technique for wider families of problems may exist.

The upper bounds of the constants for the branchwidth of planar graphs in relation to the value of the previously listed parameters, as well as for the algorithms of the dynamic programming in graphs of bounded branchwidth, emerge from a thorough inspection and they are believed to be optimized. However, any improvement to the general bound of the branchwidth in planar graphs (main result of [36], see also Theorem 5.1.1), would instantly improve the algorithmic analysis of all mentioned results.

Bibliography

[1] Karl A. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness. IV. On completeness for W[P] and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995.

[2] Noga Alon, Paul Seymour, and Robin Thomas. Planar separators. *SIAM J. Discrete Math.*, 7(2):184–193, 1994.

[3] Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *J. Artificial Intelligence Res.*, 12:219–234 (electronic), 2000.

[4] Hans L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Automata, languages and programming (Tampere, 1988)*, volume 317 of *Lecture Notes in Comput. Sci.*, pages 105–118. Springer, Berlin, 1988.

[5] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoret. Comput. Sci.*, 209(1-2):1–45, 1998.

[6] Hans L. Bodlaender. A cubic kernel for feedback vertex set. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2007)*, volume 4393 of *LNCS*, pages 320–331. Springer, Berlin, 2007.

[7] Kevin Burrage, Vladimir Estivill-Castro, Michael Fellows, Michael Langston, Shev Mac, and Frances Rosamond. The undirected feedback vertex set problem has a Poly(k) kernel. In *Parameterized and exact computation*, volume 4169 of *Lecture Notes in Comput. Sci.*, pages 192–202. Springer, Berlin, 2006.

[8] Janer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for the feedback vertex set problems. Technical Report 348, Department of Informatics, University of Bergen, Bergen, Norway, 2007.

[9] Fabián A. Chudak, Michel X. Goemans, Dorit S. Hochbaum, and David P. Williamson. A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. *Oper. Res. Lett.*, 22(4-5):111–118, 1998.

[10] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensional parameters and local treewidth. *SIAM J. Discrete Math.*, 18(3):501–511 (electronic), 2004/05.

[11] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893 (electronic), 2005.

[12] Erik D. Demaine and MohammadTaghi Hajiaghayi. Equivalence of local treewidth and linear local treewidth and its algorithmic applications. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 840–849 (electronic), New York, 2004. ACM.

[13] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. *Algorithmica*, 41(4):245–267, 2005.

[14] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. The bidimensional theory of bounded-genus graphs. *SIAM J. Discrete Math.*, 20(2):357–371 (electronic), 2006.

[15] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

[16] Frederic Dorn. *Designing Subexponential Algorithms : Problems, Techniques & Structures*. PhD thesis, Department of Informatics, University of Bergen, 2007.

[17] Frederic Dorn, Fedor V. Fomin, and Dimitrios M. Thilikos. Fast subexponential algorithm for non-local problems on graphs of bounded genus. In *Algorithm theory—SWAT 2006*, volume 4059 of *Lecture Notes in Comput. Sci.*, pages 172–183. Springer, Berlin, 2006.

[18] Frederic Dorn, Eelko Penninkx, Hans L. Bodlaender, and Fedor V. Fomin. Efficient exact algorithms on planar graphs: exploiting sphere cut branch decompositions. In *Algorithms—ESA 2005*, volume 3669 of *Lecture Notes in Comput. Sci.*, pages 95–106. Springer, Berlin, 2005.

[19] Frederic Dorn and Jan Arne Telle. Two birds with one stone: the best of branchwidth and treewidth with one algorithm. In *LATIN 2006: Theoretical informatics*, volume 3887 of *Lecture Notes in Comput. Sci.*, pages 386–397. Springer, Berlin, 2006.

[20] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.

[21] Rod Downey and Michael Fellows. Fixed-parameter tractability and completeness. III. Some structural aspects of the W hierarchy. In *Complexity theory*, pages 191–225. Cambridge Univ. Press, Cambridge, 1993.

[22] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness. In *Proceedings of the Twenty-first Manitoba Conference on Numerical Mathematics and Computing (Winnipeg, MB, 1991)*, volume 87, pages 161–178, 1992.

[23] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness. I. Basic results. *SIAM J. Comput.*, 24(4):873–921, 1995.

[24] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141(1-2):109–131, 1995.

[25] Rodney G. Downey and Michael R. Fellows. Parameterized computational feasibility. In *Feasible mathematics, II (Ithaca, NY, 1992)*, volume 13 of *Progr. Comput. Sci. Appl. Logic*, pages 219–244. Birkhäuser Boston, Boston, MA, 1995.

[26] Rodney G. Downey and Michael R. Fellows. Parameterized complexity after (almost) ten years: review and open questions. In *Combinatorics, computation & logic '99 (Auckland)*, volume 21 of *Aust. Comput. Sci. Commun.*, pages 1–33. Springer, Singapore, 1999.

[27] Rodney G. Downey, Michael R. Fellows, and Ulrike Stege. Computational tractability: the view from Mars. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, (69):73–97, 1999.

[28] Rodney G. Downey, Michael R. Fellows, and Ulrike Stege. Parameterized complexity: a framework for systematically confronting computational intractability. In *Contemporary trends in discrete mathematics (Štiřín Castle, 1997)*, volume 49 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 49–99. Amer. Math. Soc., Providence, RI, 1999.

[29] Michael R. Fellows. Parameterized complexity: The main ideas and connections to practical computing. In *Experimental Algorithmics*, volume 2547 of *Lecture Notes in Comput. Sci.*, pages 51–77. Springer, Berlin, 2002.

[30] Michael R. Fellows. New directions and new challenges in algorithm design and complexity, parameterized. In *Algorithms and data structures*, volume 2748 of *Lecture Notes in Comput. Sci.*, pages 505–519. Springer, Berlin, 2003.

[31] Henning Fernau and David Juedes. A geometric approach to parameterized algorithms for domination problems on planar graphs. In *Mathematical foundations of computer science 2004*, volume 3153 of *Lecture Notes in Comput. Sci.*, pages 488–499. Springer, Berlin, 2004.

[32] Paola Festa, Panos M. Pardalos, and Mauricio G. C. Resende. Feedback set problems. In *Handbook of combinatorial optimization, Supplement Vol. A*, pages 209–258. Kluwer Acad. Publ., Dordrecht, 1999.

[33] J. Flum and M. Grohe. *Parameterized complexity theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.

[34] Fedor V. Fomin, Serge Gaspers, Artem V. Pyatkin, and Igor Razgon. On the minimum feedback vertex set problem: Exact and enumeration algorithms. *Algorithmica*, To appear, 2008.

[35] Fedor V. Fomin and Dimitrios M. Thilikos. Dominating sets in planar graphs: branch-width and exponential speed-up. *SIAM J. Comput.*, 36(2):281–309 (electronic), 2006.

[36] Fedor V. Fomin and Dimitrios M. Thilikos. New upper bounds on the decomposability of planar graphs. *Journal of Graph Theory*, 51(1):53–81, 2006.

[37] Michael R. Garey and David S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* W. H. Freeman and Co., San Francisco, Calif., 1979.

[38] Michel X. Goemans and David P. Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. In *Integer programming and combinatorial optimization (Vancouver, BC, 1996)*, volume 1084 of *Lecture Notes in Comput. Sci.*, pages 147–161. Springer, Berlin, 1996.

[39] Michel X. Goemans and David P. Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica*, 18(1):37–59, 1998.

[40] Qian-Ping Gu and Hisao Tamaki. Optimal branch-decomposition of planar graphs in $O(n^3)$ time. In *Automata, languages and programming*, volume 3580 of *Lecture Notes in Comput. Sci.*, pages 373–384. Springer, Berlin, 2005.

[41] Ton Kloks, C. M. Lee, and Jiping Liu. New algorithms for k-face cover, k-feedback vertex set, and k-disjoint set on plane and planar graphs. In *The 28th International Workshop on Graph-Theoretic Concepts in Computer Science(WG 2002)*, page to appear. Springer, Lecture Notes in Computer Science, Berlin, 2002.

[42] Hen-Ming Lin and Jing-Yang Jou. On computing the minimum feedback vertex set of a directed graph by contraction operations. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 19(3):295–307, Mar 2000.

[43] Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications.* Oxford University Press, Oxford, 2006.

[44] Neil Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.

[45] Neil Robertson and P. D. Seymour. Graph minors. XI. Circuits on a surface. *J. Combin. Theory Ser. B*, 60(1):72–106, 1994.

[46] Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.