The Bidimensionality Theory and Its Algorithmic Applications

by

MohammadTaghi Hajiaghayi

B.S., Sharif University of Technology, 2000
M.S., University of Waterloo, 2001

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author ....
Department of Mathematics
April 29, 2005

Certified by ....
Erik D. Demaine
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Rodolfo Ruben Rosales
Chairman, Applied Mathematics Committee

Accepted by ....
Pavel I. Etingof
Chairman, Department Committee on Graduate Students

The Bidimensionality Theory and Its Algorithmic Applications

by

MohammadTaghi Hajiaghayi

Submitted to the Department of Mathematics
on April 29, 2005, in partial fulfillment of the
requirements for the degree of
DOCTOR OF PHILOSOPHY

Abstract

Our newly developing theory of bidimensional graph problems provides general techniques for designing efficient fixed-parameter algorithms and approximation algorithms for NP-hard graph problems in broad classes of graphs. This theory applies to graph problems that are *bidimensional* in the sense that (1) the solution value for the $k \times k$ grid graph (and similar graphs) grows with k, typically as $\Omega(k^2)$, and (2) the solution value goes down when contracting edges and optionally when deleting edges. Examples of such problems include feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal parameters, dominating set, edge dominating set, r-dominating set, connected dominating set, connected edge dominating set, connected r-dominating set, and unweighted TSP tour (a walk in the graph visiting all vertices). Bidimensional problems have many structural properties; for example, any graph embeddable in a surface of bounded genus has treewidth bounded above by the square root of the problem's solution value. These properties lead to efficient—often subexponential—fixed-parameter algorithms, as well as polynomial-time approximation schemes, for many minor-closed graph classes. One type of minor-closed graph class of particular relevance has bounded local treewidth, in the sense that the treewidth of a graph is bounded above in terms of the diameter; indeed, we show that such a bound is always at most linear. The bidimensionality theory unifies and improves several previous results. The theory is based on algorithmic and combinatorial extensions to parts of the Robertson-Seymour Graph Minor Theory, in particular initiating a parallel theory of graph contractions. The foundation of this work is the topological theory of drawings of graphs on surfaces and our results regarding the relation (the linearity) of the size of the largest grid minor in terms of treewidth in bounded-genus graphs and more generally in graphs excluding a fixed graph H as a minor.

In this thesis, we also develop the algorithmic theory of vertex separators, and its relation to the embeddings of certain metric spaces. Unlike in the edge case, we show that embeddings into L_1 (and even Euclidean embeddings) are insufficient, but that the additional structure provided by many embedding theorems does suffice for our purposes. We obtain an $O(\sqrt{\log n})$ approximation for min-ratio vertex cuts in general graphs, based on a new semidefinite relaxation of the problem, and a tight analysis of the integrality gap which is shown to be $\Theta(\sqrt{\log n})$. We also prove various approximate max-flow/min-vertex-

cut theorems, which in particular give a constant-factor approximation for min-ratio vertex cuts in any excluded-minor family of graphs. Previously, this was known only for planar graphs, and for general excluded-minor families the best-known ratio was $O(\log n)$. These results have a number of applications. We exhibit an $O(\sqrt{\log n})$ pseudo-approximation for finding balanced vertex separators in general graphs. Furthermore, we obtain improved approximation ratios for treewidth: In any graph of treewidth k, we show how to find a tree decomposition of width at most $O(k\sqrt{\log k})$, whereas previous algorithms yielded $O(k \log k)$. For graphs excluding a fixed graph as a minor, we give a constant-factor approximation for the treewidth; this via the bidimensionality theory can be used to obtain the first polynomial-time approximation schemes for problems like minimum feedback vertex set and minimum connected dominating set in such graphs.

Thesis Supervisor: Erik D. Demaine
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I am deeply indebted to my thesis advisor, Professor Erik Demaine, and my academic advisor, Professor Tom Leighton, for being enthusiastic and genius supervisors. Erik and I were more friends than advisor and advisee and we spent countless hours together exchanging puzzles, research ideas, and philosophies. Between us, we have explored lots of plausible ideas in algorithm design, many two or more times. His insights (technical and otherwise) have been invaluable to me. Erik also helped me a lot with his useful suggestions on the presentation of the results and the writing style. I am grateful to Tom for his perceptiveness, and his deep insights over the years on my research. He provided me a great source of inspiration. Both Erik and Tom have been extremely generous in giving me a great deal of their valuable time and sharing with me their ideas and insights. This research would have been impossible without their helps and encouragements.

I thank Professor Uriel Feige, Professor Fedor Fomin, Professor Naomi Nishimura, Professor Prabhakar Ragde, Professor Paul Seymour, Professor Dan Spielman, Professor Dimitrios Thilikos, and James Lee for fruitful collaborations and discussions in the areas related to this thesis. I especially thank Paul Seymour for many helpful discussions and for providing a portal into the Graph Minor Theory and revealing some of its hidden structure that we use in this thesis. Thanks go to Professor Daniel Kleitman who served on my committee, read my thesis, and gave me useful comments. Furthermore, I would like to thank the staff and the faculty of the Department of Mathematics and Computer Science and Artificial Intelligence Laboratory at MIT, especially Kathleen Dickey and Linda Okun, for providing such a nice academic environment. I am also grateful to the researchers at IBM T.J. Watson Research Center and Microsoft Research for two great summers in the research industry.

My friends here at MIT, namely Reza Alam, Mihai Badoiu, Saeed Bagheri, Mohsen Bahramgiri, Eaman Eftekhary, Nick Harvey, Fardad Hashemi, Susan Hohenberger, Nicole Immorlica, Ali Khakifirooz, Bobby Kleinberg, Mahnaz Maddah, Mahammad Mahdian, Vahab Mirrokni, Eddie Nikolova, Hazhir Rahmandad, Mohsen

Razavi, Navid Sabbaghi, Saeed Saremi, Anastasios Sidiropoulos, Ali Tabaei, Mana Taghdiri, David Woodruff, Sergey Yekhanin, and many others played important roles in making my life more enjoyable. Also, I would like to extend my appreciation to thank my previous advisors in University of Waterloo, namely Professor Naomi Nishimura, and in Sharif University of Technology, namely Professor Ebad Mahmoodian and Professor Mohammad Ghodsi, and my many other friends in Iran, Canada and USA for their warmest friendship. I am blessed to have such excellent companies.

Last but not least, my dear parents and my family receive my heartfelt gratitude for their sweetest support and never-ending love. I always feel the warmth of their love, even now that we are so far away. I wish to thank my brother, Mahdi, and my sisters, Monir and Mehri, for being the greatest source of love. This thesis is dedicated to my parents and my family.

Contents

List of Figures

Chapter 1

Introduction and Overview

The newly developing theory of bidimensional graph problems, developed in a series of papers [74, 72, 64, 65, 63, 66, 62, 73, 71, 70], provides general techniques for designing efficient fixed-parameter algorithms and approximation algorithms for NP-hard graph problems in broad classes of graphs. This theory applies to graph problems that are *bidimensional* in the sense that (1) the solution value for the $k \times k$ grid graph (and similar graphs) grows with k, typically as $\Omega(k^2)$, and (2) the solution value goes down when contracting edges and optionally when deleting edges. Examples of such problems include feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal parameters, dominating set, edge dominating set, R-dominating set, connected dominating set, connected edge dominating set, connected R-dominating set, and unweighted TSP tour (a walk in the graph visiting all vertices).

Bidimensional problems have many structural properties; for example, any graph in an appropriate minor-closed class has treewidth bounded above in terms of the problem's solution value, typically by the square root of that value. These properties lead to efficient—often subexponential—fixed-parameter algorithms, as well as polynomial-time approximation schemes, for many minor-closed graph classes. One type of minor-closed graph class of particular relevance has *bounded local treewidth*, in the sense that the treewidth of a graph is bounded above in terms of the diameter; indeed, such a bound is always at most linear.

The bidimensionality theory unifies and improves several previous results. The theory is based on algorithmic and combinatorial extensions to parts of the Robertson-Seymour Graph Minor Theory, in particular initiating a parallel theory of graph contractions. The foundation of this work is the topological theory of drawings of graphs on surfaces.

This chapter is organized as follows. We start with our graph terminology in Section 1.1. Section 1.2 defines the various graph classes of increasing generality to which bidimensionality theory applies. Section 1.3 describes several structural properties of graphs in these classes, in particular from Graph Minor Theory, that form the basis of bidimensionality. Section 1.4 defines bidimensional parameters and problems and gives some examples. Section 1.5 describes one of the main structural properties of bidimensionality, namely, that the treewidth is bounded in terms of the parameter value. Sections 1.6–1.11 describe several consequences of bidimensionality theory: separator theorems, bounds on local treewidth, fixed-parameter algorithms, and polynomial-time approximation schemes. Finally in Section 1.12, we describe the structure of this thesis.

1.1 Graph Terminology

We assume the reader is familiar with general concepts of graph theory such as graphs, trees, and planar graphs. The reader is referred to standard references for appropriate background [40]. For exact definitions of various NP-hard problems in this paper, the reader is referred to Garey and Johnson's seminal book [99]. Here we review a few terms used in this thesis; others will be defined in the context of their use.

Throughout this thesis, all graphs are finite, simple, and undirected, unless indicated otherwise. A graph G is represented by $G = (V, E)$, where V (or $V(G)$) is the set of vertices and E (or $E(G)$) is the set of edges; we use n to denote $|V|$ when G is clear from context. An edge e in a graph G between u and v is denoted by $\{u, v\}$ or, equivalently, $\{v, u\}$. Here, vertices u and v are called the *endpoints* of e. A graph $G' = (V', E')$ is a *subgraph of* G if $V' \subseteq V$ and $E' \subseteq E$, and is an *induced subgraph of* G, denoted by $G[V']$, if in addition E' contains all edges of E that have both endpoints in V'. The (disjoint) *union* of two disjoint graphs G_1 and G_2, $G_1 \cup G_2$, is a graph G formed by merging vertex and edge sets, so that $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2)$.

We define the *r-neighborhood* of a set $S \subseteq V(G)$, denoted by $N_G^r(S)$, to be the set of vertices at distance at most r from at least one vertex of $S \subseteq V(G)$; if $r = 1$, we simply use the notation $N_G(S)$ and if $S = \{v\}$, we simply use the notation $N_G^r(v)$. We also define $N_G[v] = N_G(v) - \{v\}$. The *diameter* of G, denoted by $diam(G)$, is the maximum over all distances between pairs of vertices of G. An *n-clique (K_n)* is an n-vertex graph in which every pair of vertices is connected by an edge. The vertices of the graph $K_{n,m}$ can be partitioned into sets V_1 and V_2 such that $|V_1| = n$, $|V_2| = m$, and the edge set consists of all edges $\{u, v\}$ such that $u \in V_1$ and $v \in V_2$. A graph $G = (V, E)$ is *k-connected* if for any $S \subseteq V(G)$ such that $|S| \leq k$, $G[V - S]$ is connected.

1.2 Graph Classes

In this section, we introduce several families of graphs, each playing an important role in both the Graph Minor Theory and the bidimensionality theory. Refer to Figure 1-1. All of these graph classes are generalizations of planar graphs, which are well-studied in algorithmic graph theory. Unlike planar graphs and map graphs, every other class of graphs we consider can include any particular graph G; of course, this inclusion requires a bound or excluded minor large enough depending on G. This property distinguishes this line of research from other work considering exclusion of particular minors, e.g., $K_{3,3}$, K_5, or K_6.

1.2.1 Definitions of Graph Classes

The first three classes of graphs relate to embeddings on surfaces. A graph is *planar* if it can be drawn in the plane (or the sphere) without crossings. A graph has *genus*

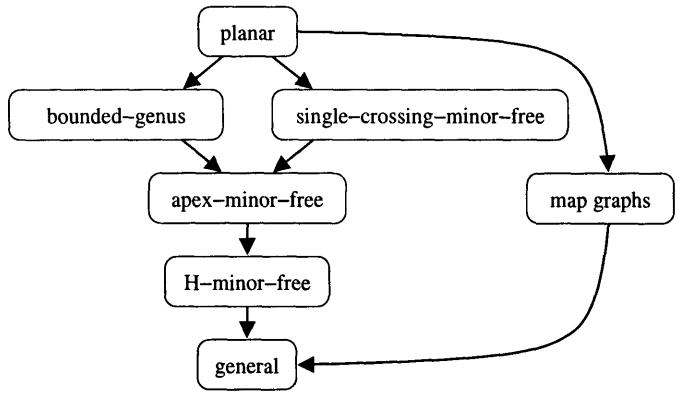

Figure 1-1: Interesting classes of graphs. Arrows point from more specific classes to more inclusive classes.

at most g if it can be drawn in an orientable surface of genus g without crossings.[1] A class of graphs has *bounded genus* if every graph in the class has genus at most g for a fixed g.

Given an embedded planar graph and a two-coloring of its faces as either *nations* or *lakes*, the associated *map graph* has a vertex for each nation and an edge between two vertices corresponding to nations (faces) that share a vertex. The dual graph is defined similarly, but with adjacency requiring a shared edge instead of just a shared vertex. Map graphs were introduced by Chen, Grigni, and Papadimitriou [58] as a generalization of planar graphs that can have arbitrarily large cliques. Thorup [162] gave a polynomial-time algorithm for constructing the underlying embedded planar graph and face two-coloring for a given map graph, or determining that the given graph is not a map graph.

We view the class of map graphs as a special case of taking fixed powers of a family of graphs. The *kth power* G^k of a graph G is the graph on the same vertex set $V(G)$ with edges connecting two vertices in G^k precisely if the distance between these vertices in G is at most k. For a bipartite graph G with bipartition $V(G) = U \cup W$, the *half-square* $G^2[U]$ is the graph on one side U of the partition, with two vertices adjacent in $G^2[U]$ precisely if the distance between these vertices in G is 2. A graph is

[1]This definition also includes graphs that can be drawn in non-orientable surfaces of low genus, because if a graph has non-orientable genus g, then it has orientable genus at most $2g$.

a map graph if and only if it is the half-square of some planar bipartite graph [58]. In fact, this translation between map graphs and half-squares is constructive and takes polynomial time.

The next three classes of graphs relate to excluding minors. Given an edge $e = \{v, w\}$ in a graph G, the *contraction* of e in G is the result of identifying vertices v and w in G and removing all loops and duplicate edges (the resulting graph is denoted by G/e). A graph H obtained by a sequence of such edge contractions starting from G is said to be a *contraction* of G. A graph H is a *minor* of G if H is a subgraph of some contraction of G. We use the notation $H \preceq G$ (resp. $H \preceq_c G$) for H is a minor (a contraction) of G. A graph class $\mathcal{C}$ is *minor-closed* if any minor of any graph in $\mathcal{C}$ is also a member of $\mathcal{C}$. A minor-closed graph class $\mathcal{C}$ is *H-minor-free* if $H \notin \mathcal{C}$. More generally, we use the term "H-minor-free" to refer to any minor-closed graph class that excludes some fixed graph H.

A *single-crossing graph* is a minor of a graph that can be drawn in the plane with at most one pair of edges crossing. Note that a single-crossing graph may not itself be drawable with at most one crossing pair of edges; see Section 2.1. Such graphs were first defined by Robertson and Seymour [142]. A minor-closed graph class is *single-crossing-minor-free* if it excludes a fixed single-crossing graph.

An *apex graph* is a graph in which the removal of some vertex leaves a planar graph. A graph class is *apex-minor-free* if it excludes some fixed apex graph. Such graph classes were first considered by Eppstein [85, 87], in connection to the notion of bounded local treewidth as described in Section 1.7.

The next section describes strong structural properties of the last three classes of graphs (minor-excluding classes) in terms of the first two classes of graphs (embeddable on surfaces) and other ingredients.

1.3 Structural Properties

Graphs from single-crossing-minor-free and H-minor-free graph classes have powerful structural properties from the Graph Minor Theory. First we need to define treewidth, pathwidth, and clique sums.

1.3.1 Background

Treewidth: Many difficult graph problems can be solved efficiently when the input is restricted to graphs of bounded treewidth (see e.g., Bodlaender's survey [31]). The notion of treewidth first was introduced by Robertson and Seymour [143]. To define this notion, first we consider a representation of a graph as a tree, called a tree decomposition. Precisely, a *tree decomposition* of a graph $G = (V, E)$ is a pair (T, χ) in which $T = (I, F)$ is a tree and $\chi = \{\chi_i \mid i \in I\}$ is a family of subsets of $V(G)$ such that

1. $\bigcup_{i \in I} \chi_i = V$;

2. for each edge $e = \{u, v\} \in E$, there exists an $i \in I$ such that both u and v belong to χ_i; and

3. for all $v \in V$, the set of nodes $\{i \in I \mid v \in \chi_i\}$ forms a connected subtree of T.

To distinguish between vertices of the original graph G and vertices of T in the tree decomposition, we call vertices of T *nodes* and their corresponding χ_i's *bags*. The *width* of the tree decomposition is the maximum size of a bag in χ minus 1. The *treewidth* of a graph G, denoted by $\mathbf{tw}(G)$, is the minimum width over all possible tree decompositions of G. A tree decomposition is called a *path decomposition* if $T = (I, F)$ is a path. The *pathwidth* of a graph G, denoted $\mathbf{pw}(G)$, is the minimum width over all possible path decompositions of G.

A *graph of bounded treewidth* is a graph of treewidth at most k, for k a constant independent of the size of the graph. A related notion is that of a *k-tree* [152], a graph G such that either G is a k-clique or G has a vertex u of degree k such that u is adjacent to a k-clique, and the graph obtained by deleting u and all its incident edges is a k-tree. It has been shown that for any k, the class of graphs of treewidth at most k is equivalent to the class of *partial k-trees*, that is, subgraphs of k-trees [127].

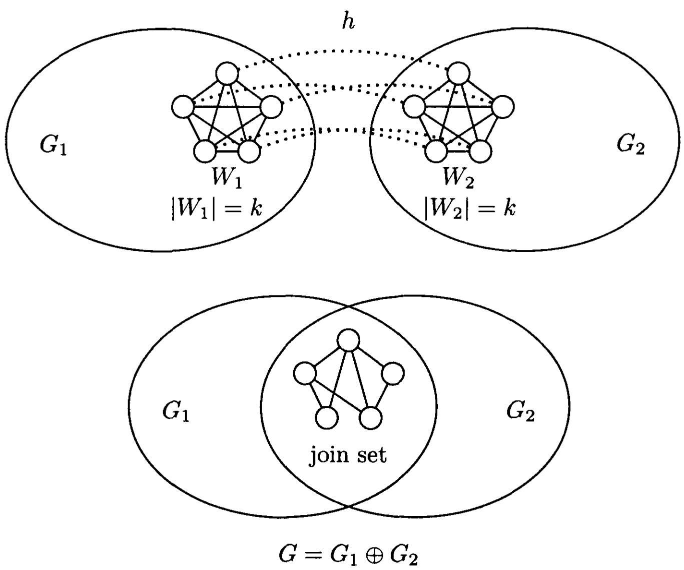

Figure 1-2: Example of a 5-sum of two graphs.

Clique sum: The notion of clique sums goes back to characterizations of $K_{3,3}$-minor-free and K_5-minor-free graphs by Wagner [164] and serves as an important tool in the Graph Minor Theory. Suppose G_1 and G_2 are graphs with disjoint vertex sets and let $k \geq 0$ be an integer. For $i = 1, 2$, let $W_i \subseteq V(G_i)$ form a clique of size k and let G'_i be obtained from G_i by deleting some (possibly no) edges from the induced subgraph $G_i[W_i]$ with both endpoints in W_i. Consider a bijection $h : W_1 \to W_2$. We define a *k-sum* G of G_1 and G_2, denoted by $G = G_1 \oplus_k G_2$ or simply by $G = G_1 \oplus G_2$, to be the graph obtained from the union of G'_1 and G'_2 by identifying w with $h(w)$ for all $w \in W_1$. The images of the vertices of W_1 and W_2 in $G_1 \oplus_k G_2$ form the *join set*.

Note that each vertex v of G has a corresponding vertex in G_1 or G_2 or both. It is also worth mentioning that $\oplus$ is not a well-defined operator: it can have a set of possible results. More specifically, the result of $\oplus$ will depend on which (if any) edges are removed from the cliques as well as which bijection is selected, so the operation $\oplus$ can have a set of possible results, and hence is not well-defined. A series of k-sums (not necessarily unique) that generate a graph G is called *a decomposition of G into clique-sum operations.*

Figure 1-2 demonstrates an example of a 5-sum operation.

The reader is referred to [77] to see more results on clique-sum classes.

1.3.2 Structure of Single-Crossing-Minor-Free Graphs

The structure of single-crossing-minor-free graphs can be described as follows:

Theorem 1.1 ([142]). *For any fixed single-crossing graph H, every H-minor-free graph can be obtained by a sequence of k-sums, $0 \le k \le 3$, of planar graphs and graphs of bounded treewidth, where the bound on treewidth depends on H.*

This theorem generalizes characterizations of $K_{3,3}$-minor-free and K_5-minor-free graphs [164]. A graph is $K_{3,3}$-minor-free if and only if it can be obtained by k-sums, $0 \le k \le 2$, of planar graphs and K_5. A graph is K_5-minor-free if and only if it can be obtained by k-sums, $0 \le k \le 3$, of planar graphs and V_8 (the length-8 cycle C_8 together with eight edges joining diametrically opposite vertices).

This structural property of single-crossing-minor-free graphs has since been strengthened to ensure that the summands are minors of the original graph and to provide algorithms for finding the decomposition:

Theorem 1.2 ([72], see also Chapter 2). *For any fixed single-crossing graph H, there is an $O(n^4)$-time algorithm to compute, given an H-minor-free graph G, a decomposition of G as a sequence of k-sums, $0 \le k \le 3$, of planar graphs and graphs of bounded treewidth (where the bound on treewidth depends on H), each of which is a minor of G.*

1.3.3 Structure of H-Minor-Free Graphs

The structure of H-minor-free graphs is described by a deep theorem of Robertson and Seymour [149]. Intuitively, their theorem says that, for every graph H, every H-minor-free graph can be expressed as a "tree structure" of pieces, where each piece is a graph that can be drawn in a surface in which H cannot be drawn, except for a bounded number of "apex" vertices and a bounded number of "local areas of non-planarity" called *vortices*. Here the bounds depend only on H.

Roughly speaking, we say that a graph G is *h-almost embeddable* in a surface S if there exists a set X of size at most h of vertices, called *apex vertices* or *apices*, such that $G - X$ can be obtained from a graph G_0 embedded in S by attaching at most h graphs of pathwidth at most h to G_0 within h faces in an orderly way. More precisely, a graph G is *h-almost embeddable* in S if there exists a vertex set X of size at most h (the *apices*) such that $G - X$ can be written as $G_0 \cup G_1 \cup \cdots \cup G_h$, where

1. G_0 has an embedding in S;
2. the graphs G_i, called *vortices*, are pairwise disjoint;
3. there are faces $F_1, \ldots, F_h$ of G_0 in S, and there are pairwise disjoint disks $D_1, \ldots, D_h$ in S, such that for $i = 1, \ldots, h$, $D_i \subset F_i$ and $U_i := V(G_0) \cap V(G_i) = V(G_0) \cap D_i$; and
4. the graph G_i has a path decomposition $(\mathcal{B}_u)_{u \in U_i}$ of width less than h, such that $u \in \mathcal{B}_u$ for all $u \in U_i$. The sets $\mathcal{B}_u$ are ordered by the ordering of their indices u as points along the boundary cycle of face F_i in G_0.

An h-almost embeddable graph is *apex-free* if the set X of apices is empty.

Now, the deep result of Robertson and Seymour is as follows:

Theorem 1.3 ([149]). *For every graph H, there exists an integer $h \geq 0$ depending only on $|V(H)|$ such that every H-minor-free graph can be obtained by at most h-sums of graphs that are h-almost-embeddable in some surfaces in which H cannot be embedded.*

In particular, if H is fixed, any surface in which H cannot be embedded has bounded genus. Thus, the summands in the theorem are h-almost-embeddable in bounded-genus surfaces.

Another way to view Theorem 1.3 is that every H-minor-free graph G has a tree decomposition (T, χ) such that, for each node $i \in V(T)$, the induced subgraph $G[\chi_i]$ augmented with additional edges to form a clique on the vertices that overlap with the parent's bag, and a clique on the vertices that overlap with each child's bag, is h-almost-embeddable in a bounded-genus surface. (This augmented graph is called the *torso* $[\chi_i]$ in, e.g., [103, 81].) The intersections between bag χ_i and its parent's bag, and with each child's bag, correspond to the join sets in the clique-sum decomposition. Our development primarily follows the original clique-sum viewpoint of Robertson and Seymour, but we will also occasionally view the sums as being organized into the tree T.

Theorem 1.3 is very general and appeared in print only recently. However already several nice applications (see e.g. [39, 103, 75]) are known.

As observed by Seymour [154], the constructive proof of Theorem 1.3 in [149] also establishes the following algorithmic result. (Grohe [103] also shows how to obtain a similar result using Robertson and Seymour's theorem that every minor-closed class of graphs has a polynomial-time membership test [149].)

Theorem 1.4. [149, 154] *For any graph H, there is an algorithm with running time $n^{O(1)}$ that either computes a clique-sum decomposition as in Theorem 1.3 for any given H-minor-free graph G, or outputs that G is not H-minor-free. The exponent in the running time depends on H.*

1.3.4 Structure of Apex-Minor-Free Graphs

Apex-minor-free graph classes are an important subfamily of H-minor-free graph classes. The general structural theorem for H-minor-free graphs applies in this context as well. However, reductions developed in [66] suggest that the decomposition can be restricted to a particular form in the apex-minor-free case:

Conjecture 1.5 ([66]). *For every graph H, there is an integer $h \geq 0$ depending only on $|V(H)|$ such that every H-minor-free graph can be obtained by at most h-sums of graphs that are h-almost-embeddable in some surfaces in which H cannot be embedded and whose apices are connected via edges only to vertices within vortices.*

1.3.5 Grid Minors

The $r \times r$ grid is the canonical planar graph of treewidth $\Theta(r)$. In particular, an important result of Robertson, Seymour, and Thomas [151] is that every planar graph of treewidth w has an $\Omega(w) \times \Omega(w)$ grid graph as a minor. Thus every planar graph of large treewidth has a grid minor certifying that its treewidth is almost as large (up to constant factors). Recently, this result has been generalized to any H-minor-free graph class:

Theorem 1.6 ([71], see also Chapter 8). *For any fixed graph H, every H-minor-free graph of treewidth w has an $\Omega(w) \times \Omega(w)$ grid as a minor.*

Thus the $r \times r$ grid is the canonical H-minor-free graph of treewidth $\Theta(r)$ for any fixed graph H. This result is also best possible up to constant factors. Chapter 8 discusses the remaining issue of bounding the constant factor and its dependence on H.

A similar but weaker bound plays an important role in the Graph Minor Theory [144]: for any fixed graph H and integer $r > 0$, there is an integer $w > 0$ such that every H-minor-free graph with treewidth at least w has an $r \times r$ grid graph as a minor. This result has been re-proved by Robertson, Seymour, and Thomas [151], Reed [141], and Diestel, Jensen, Gorbunov, and Thomassen [80]. Among these proofs, the best known bound on w in terms of r is that every H-minor-free graph of treewidth larger than $20^{5|V(H)|^3 r}$ has an $r \times r$ grid as a minor [151]. Theorem 1.6 therefore offers an exponential (and best possible) improvement over previous results.

Theorem 1.6 cannot be generalized to arbitrary graphs: Robertson, Seymour, and Thomas [151] proved that some graphs have treewidth $\Omega(r^2 \lg r)$ but have grid minors only of size $O(r) \times O(r)$. The best known relation for general graphs is that having treewidth more than 20^{2r^5} implies the existence of an $r \times r$ grid minor [151]. The best possible bound is believed to be closer to $\Theta(r^2 \lg r)$ than $2^{\Theta(r^5)}$, perhaps even equal to $\Theta(r^2 \lg r)$ [151]. In fact, we [69] conjecture that the correct bound is $\Theta(r^3)$.

1.4 Bidimensional Parameters/Problems

Bidimensionality has been introduced and developed in a series of papers [74, 72, 64, 65, 63, 66, 62, 73, 71, 70]. Although implicitly hinted at in [74, 72, 64, 65], the first use of the term "bidimensional" was in [63].

First we define "parameters" as an alternative view on optimization problems. A *graph parameter* P (or just a *parameter* P, when it clear in the context) is a function mapping graphs to nonnegative integers. The *decision problem associated with* P asks, for a given graph G and nonnegative integer k, whether $P(G) \leq k$. Many optimization problems can be phrased as such a decision problem about a graph parameter P.

Now we can define bidimensionality. A parameter is $g(r)$-*bidimensional* (or just *bidimensional*) if it is at least $g(r)$ in an $r \times r$ "grid-like graph" and if the parameter does not increase when taking either minors ($g(r)$-*minor-bidimensional*) or contractions ($g(r)$-*contraction-bidimensional*). The exact definition of "grid-like graph" depends on the class of graphs allowed and whether we are considering minor- or contraction-bidimensionality. For minor-bidimensionality and for any H-minor-free graph class, the notion of a "grid-like graph" is defined to be the $r \times r$ *grid*, i.e., the planar graph with r^2 vertices arranged on a square grid and with edges connecting horizontally and vertically adjacent vertices. For contraction-bidimensionality, the notion of a "grid-like graph" is as follows:

1. For planar graphs and single-crossing-minor-free graphs, a "grid-like graph" is an $r \times r$ grid partially triangulated by additional edges that preserve planarity.

2. For bounded-genus graphs, a "grid-like graph" is such a partially triangulated $r \times r$ grid with up to genus(G) additional edges ("handles").

3. For apex-minor-free graphs, a "grid-like graph" is an $r \times r$ grid augmented with additional edges such that each vertex is incident to $O(1)$ edges to nonboundary vertices of the grid. (Here $O(1)$ depends on the excluded apex graph.)

Contraction-bidimensionality is so far undefined for H-minor-free graphs (or general graphs).[2]

Examples of bidimensional parameters include the number of vertices, the diameter, and the size of various structures such as feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal parameters, dominating set, edge dominating set, R-dominating set, connected dominating set, connected edge dominating set, connected R-dominating set, and unweighted TSP tour (a walk in the graph visiting all vertices). (See [63, 62] for arguments of either contraction- or minor-bidimensionality for the above parameters.) We also say that the corresponding optimization problems based on these parameters, e.g., finding the minimum-size dominating set, are *bidimensional.* With the exception of diameter, all of these bidimensional problems are $\Theta(r^2)$-bidimensional, which is the most common case (and in some papers used as the definition of bidimensionality). Diameter is the main exception, being only $\Theta(r)$-contraction-bidimensional for planar graphs, single-crossing-minor-free graphs, and bounded-genus graphs, and only $\Theta(\lg r)$-contraction-bidimensional for apex-minor-free graphs.

1.5 Parameter-Treewidth Bounds

The genesis of bidimensionality was in fact the notion of a parameter-treewidth bound. A *parameter-treewidth bound* is an upper bound $f(P(G))$ on the treewidth of a graph with parameter P. Given a graph parameter P, we say that a graph family $\mathcal{F}$ has the *parameter-treewidth property for P* if there is a strictly increasing function f such that every graph $G \in \mathcal{F}$ has treewidth at most $f(P(G))$. In many cases, $f(k)$ can even be shown to be sublinear in k, often $O(\sqrt{k})$, where $k = P(G)$ for a graph G. Parameter-treewidth bounds have been established for many parameters and graph classes; see, e.g., [2, 116, 93, 6, 50, 123, 107, 64, 72, 74, 62, 66, 63]. Essentially all of these bounds can be obtained from the theory of bidimensional parameters. Thus bidimensionality is the most powerful method so far for establishing parameter-

[2]For the parameters to which we have applied bidimensionality, contraction-bidimensionality does not seem to extend beyond apex-minor-free graphs, but perhaps a suitably extended definition could be found in the context of different applications or a "theory of graph contractions".

treewidth bounds, encompassing all such previous results for H-minor-free graphs.

The central result in bidimensionality that generalizes these bounds is that every bidimensional parameter has a parameter-treewidth bound, in its corresponding family of graphs as defined in Section 1.4. More precisely, we have the following result:

Theorem 1.7 ([71, 62], see also Chapters 8 and 7). *If the parameter P is $g(r)$-bidimensional, then for every graph G in the family associated with the parameter P, $\mathbf{tw}(G) = O(g^{-1}(P(G)))$. In particular, if $g(r) = \Theta(r^2)$, then the bound becomes $\mathbf{tw}(G) = O(\sqrt{P(G)})$.*

This theorem is based on the grid-minor bound from Theorem 1.6 and the proof of a weaker parameter-treewidth bound, $\mathbf{tw}(G) = (g^{-1}(P(G)))^{O(g^{-1}(P(G)))}$, established in [62] (see also Chapter 7). The stronger bound of $\mathbf{tw}(G) = O(g^{-1}(P(G)))$ was obtained first for planar graphs [64](see also Chapter 4), then single-crossing-minor-free graphs [74, 72](see also Chapters 2 and 3), then bounded-genus graphs [63, 73] (see also Chapter 5), and finally apex-minor-free graphs for contraction-bidimensional parameters and H-minor-free graphs for minor-bidimensional parameters [71] (Theorem 1.7 above).

We can extend the definition of $g(r)$-minor-bidimensionality to general graphs by again defining a "grid-like graph" to be the $r \times r$ grid. Still we can obtain a parameter-treewidth bound [151, 68], but the bound is weaker: $\mathbf{tw}(G) = 2^{O(g^{-1}(k))^5}$.

1.6 Separator Theorems

If we apply the parameter-treewidth bound of Theorem 1.7 to the parameter of the number of vertices in the graph, which is minor-bidimensional with $g(r) = r^2$, then we immediately obtain the following (known) bound on the treewidth of an H-minor-free graph:

Theorem 1.8 ([9, Proposition 4.5], [103, Corollary 24], [71]). *For any fixed graph H, every H-minor-free graph G has treewidth $O(\sqrt{|V(G)|})$.*

A consequence of this result is that every vertex-weighted H-minor-free graph G has a vertex separator of size $O(\sqrt{|V(G)|})$ whose removal splits the graph into two parts each with weight at most 2/3 of the original weight [9, Theorem 1.2]. This generalization of the classic planar separator theorem has many algorithmic applications; see e.g. [9, 5]. Also, this result shows that the structural properties of H-minor-free graphs given by Theorem 1.3 are powerful enough to conclude that these graphs have small separators, which we expect from such a strong theorem.

Chapter 8 discusses the issue of how tight a lead constant can be obtained in such a result.

1.7 Local Treewidth

Eppstein [87] introduced the *diameter-treewidth property* for a class of graphs, which requires that the treewidth of a graph in the class is upper bounded by a function of its diameter. He proved that a minor-closed graph family has the diameter-treewidth property precisely if the graph family excludes some apex graph. In particular, he proved that any graph in such a family with diameter D has treewidth at most $2^{2^{O(D)}}$. (A simpler proof of this result was obtained in [65] (see also Chapter 6).)

If we apply the parameter-treewidth bound of Theorem 1.7 to the diameter parameter, which is contraction-bidimensional with $g(r) = \Theta(\lg r)$ [65], then we immediately obtain the following stronger diameter-treewidth bound for apex-minor-free graphs:

Theorem 1.9 ([71], see also Chapter 8). *For any fixed apex graph H, every H-minor-free graph of diameter D has treewidth $2^{O(D)}$.*

This theorem is not the best possible. In some sense it is necessarily limited because it still does not exploit the full structure of H-minor-free graphs from Theorem 1.3. The difficulty is that, in a grid-like graph, the $O(1)$ edges from a vertex to non-boundary vertices can accumulate to make the diameter small. However, it is possible to show that, effectively, not too many vertices can have such edges. This fact comes from the property that there are a bounded number of apices in the clique sum decomposition of Theorem 1.3, and in an apex-minor-free graph, each apex cannot have more than a bounded number of edges to "distant" vertices. Based on this fact, a complicated proof establishes the following even stronger diameter-treewidth bound in apex-minor-free graphs:

Theorem 1.10 ([66]). *For any fixed apex graph H, every H-minor-free graph of diameter D has treewidth $O(D)$.*

This diameter-treewidth bound is the best possible up to constant factors. Thus this theorem establishes that, in minor-closed graph families, having any diameter-treewidth bound is equivalent to having a linear diameter-treewidth bound. As mentioned before, no minor-closed graph families beyond apex-minor-free graphs can have any diameter-treewidth bound. Theorem 1.10 is therefore the ultimate characterization of diameter-treewidth bounds in minor-closed graph families (up to constant factors).

The proof of Theorem 1.10 is the basis for Conjecture 1.5. In fact, Theorem 1.10 would not be hard to prove assuming Conjecture 1.5.

The diameter-treewidth property has been used extensively in a slightly modified form called the *bounded-local-treewidth property*, which requires that the treewidth of any connected subgraph of a graph in the class is upper bounded by a function of its diameter. For minor-closed graph families, these two properties are identical. Graphs of bounded local treewidth have many similar properties to both planar graphs and graphs of bounded treewidth, two classes of graphs on which many problems are substantially easier. In particular, Baker's approach for polynomial-time approximation schemes (PTASs) on planar graphs [23] applies to this setting. As a result, PTASs are known for hereditary maximization problems such as maximum independent set, maximum triangle matching, maximum H-matching, and maximum tile salvage; for minimization problems such as minimum vertex cover, minimum dominating set, minimum edge-dominating set; and for subgraph isomorphism for a fixed pattern [72, 87, 110]. Graphs of bounded local treewidth also admit several efficient fixed-parameter algorithms. In particular, Frick and Grohe [96] give a general framework for deciding any property expressible in first-order logic in graphs of bounded local treewidth. Theorem 1.10 substantially improves the running time of these algorithms, in particular improving the running time of the PTASs from $2^{2^{2^{O(1/\varepsilon)}}} n^{O(1)}$ to $2^{O(1/\varepsilon)} n^{O(1)}$, where n is the number of vertices in the graph.

1.8 Subexponential Fixed-Parameter Algorithms

A *fixed-parameter algorithm* is an algorithm for computing a parameter $P(G)$ of a graph G whose running time is $h(P(G))\, n^{O(1)}$ for some function h. The exponent $O(1)$ must be independent of G; thus the exponentiality of the algorithm is bounded by the parameter $P(G)$, and the dependence on n is only polynomial. A typical function h for many fixed-parameter algorithms is $h(k) = 2^{O(k)}$. In the last six years, several researchers have obtained *exponential speedups* in fixed-parameter algorithms in the sense that the h function reduces exponentially, e.g., to $2^{O(\sqrt{k})}$. For example, the first fixed-parameter algorithm for finding a dominating set of size k in planar graphs [3] has running time $O(8^k n)$; subsequently, a sequence of subexponential algorithms and improvements have been obtained, starting with running time $O(4^{6\sqrt{34k}} n)$ [2], then $O(2^{27\sqrt{k}} n)$ [116], and finally $O(2^{15.13\sqrt{k}} k + n^3 + k^4)$ [93]. Other subexponential algorithms for other domination and covering problems on planar graphs have also been obtained [2, 6, 50, 123, 107].

All subexponential fixed-parameter algorithms developed so far are based on showing a sublinear parameter-treewidth bound and then using an algorithm whose running time is singly exponential in treewidth and polynomial in problem size. As mentioned in Section 1.5, essentially all sublinear treewidth-parameter bounds proved so far can be obtained through bidimensionality. Theorem 1.7 and the techniques of [62] yield the following general result for designing subexponential fixed-parameter algorithms:

Theorem 1.11 ([71, 62], see also Chapters 8 and 7). *Consider a $g(r)$-bidimensional parameter P that can be computed on a graph G in $h(w)\, n^{O(1)}$ time given a tree* *decomposition of G of width at most w. Then there is an algorithm computing P on any graph G in P's corresponding graph class, with running time $[h(O(g^{-1}(k))) + 2^{O(g^{-1}(k))}]\, n^{O(1)}$. In particular, if $g(r) = \Theta(r^2)$ and $h(w) = 2^{o(w^2)}$, then this running time is subexponential in k.*

In particular, this result gives subexponential fixed-parameter algorithms for many bidimensional parameters, including feedback vertex set, vertex cover, minimum maximal matching, a series of vertex-removal parameters, dominating set, edge dominating set, R-dominating set, clique-transversal set, connected dominating set, connected edge dominating set, connected R-dominating set, and unweighted TSP tour.

For minor-bidimensional parameters, these algorithms apply to all H-minor-free graphs. The next section describes to what extent these algorithms can be extended to general graphs.

For contraction-bidimensional parameters, these algorithms apply to apex-minor-free graphs. On the other hand, subexponential fixed-parameter algorithms can be obtained for dominating set, which is contraction-bidimensional, on H-minor-free graphs [63] (see also Chapter 5), map graphs [64] (see also Chapter 4), and fixed powers of planar graphs (or even fixed powers of H-minor-free graphs) [64, 63]. These algorithms are necessarily more complicated than those produced from Theorem 1.11, because apex-minor-free graphs are precisely the minor-closed graph classes for which domatinating set has a parameter-treewidth bound [62] (see Chapter 7). An intriguing open question is whether these techniques can be extended to other contraction-bidimensional problems than dominating set, for fixed powers of H-minor-free graphs and/or other classes of graphs.

1.9 Fixed-Parameter Algorithms for General Graphs

As mentioned in Section 1.5, minor-bidimensionality can be defined for general graphs as well. In this section we show how the bidimensionality theory in this case leads to a general class of fixed-parameter algorithms.

A major result from the Graph Minor Theory (in particular [147, 150]) is that every minor-closed graph property is characterized by a finite set of forbidden minors. More precisely, for any property P on graphs such that a graph having property P implies that all its minors have property P, there is a finite set $\{H_1, H_2, \ldots, H_h\}$ of graphs such that a graph G has property P if and only if G does not have H_i as a minor for all $i = 1, 2, \ldots, h$. The algorithmic consequence of this result is that there exists an $O(n^3)$-time algorithm to decide any fixed minor-closed graph property, by finitely many calls to an $O(n^3)$-time minor test [147]. This consequence has been used to show the existence of polynomial-time algorithms for several graph problems, some of which were not previously known to be decidable [91].

However, all of these algorithmic results (except the minor test) are nonconstructive: we are guaranteed that efficient algorithms exist, but are not told what they are. The difficulty is that we know that a finite set of forbidden minors exists, but lack "a means of identifying the elements of the set, the cardinality of the set, or even the order of the largest graph in the set" [91]. Indeed, there is a mathematical sense in which any proof of the finite-forbidden-minors theorem must be nonconstructive [97].

We can apply these graph-minor results to prove the existence of algorithms to compute parameters, provided the parameters never increase when taking a minor. For any fixed parameter and any fixed $k \geq 0$, there is an $O(n^3)$-time algorithm that decides whether a graph has parameter value $\leq k$. Unfortunately, the existence of these algorithms does not necessarily imply the existence of a single fixed-parameter algorithm that works for all $k \geq 0$, because the algorithms for individual k (in particular the set of forbidden minors) might be uncomputable. We do not even know an upper bound on the running time of these algorithms as a function of n and k, because we do not know the dependence of the size of the forbidden minors on k.

In [68], fixed-parameter algorithms are constructed for nearly all parameters that never increase when taking a minor, with explicit time bounds in terms of n and k. Essentially, by assuming a few very common properties of the parameter, we obtain the generalized form of minor-bidimensionality.

Theorem 1.12 ([68]). *Consider a parameter P that is positive on some $g \times g$ grid,* *never increases when taking minors, is at least the sum over the connected components of a disconnected graph, and can be computed in $h(w)\, n^{O(1)}$ time given a width-w tree decomposition of the graph. Then there is an algorithm that decides whether P is at most k on a graph with n vertices in $\left[2^{2^{O(g\sqrt{k})^5}} + h(2^{O(g\sqrt{k})^5})\right] n^{O(1)}$ time.*

As mentioned in [68], a conjecture of Robertson, Seymour, and Thomas [151] would improve the running time to $h(O(k \lg k))\, n^{O(1)}$, which is $2^{O(k \lg k)} n^{O(1)}$ for the typical case of $h(w) = 2^{O(w)}$. This conjectured time bound almost matches the fastest known fixed-parameter algorithms for several parameters, e.g., feedback vertex set, vertex cover, and a general family of vertex-removal problems [91].

1.10 Polynomial-Time Approximation Schemes

Recently, the bidimensionality theory has been extended to obtain polynomial-time approximation schemes (PTASs) for essentially all bidimensional parameters, including those mentioned above [70]. These PTASs are based on techniques that generalize and in some sense unify the two main previous approaches for designing PTASs in planar graphs, namely, the Lipton-Tarjan separator approach [132] and the Baker layerwise decomposition approach [23]. The PTASs apply to H-minor-free graphs for minor-bidimensional parameters and to apex-minor-free graphs for contraction-bidimensional parameters. To achieve this level of generality, [70] uses the sublinear parameter-treewidth bound of Theorem 1.7 as well as an $O(1)$-approximation algorithm for treewidth in H-minor-free graphs [90] (see also Chapter 9).

Before we can state the general theorem for constructing PTASs, we need to define a few straightforward required conditions, which are commonly satisfied by most bidimensional problems. The theorem considers families of problems in which we are given a graph and our goal is to find a minimum-size set of vertices and/or edges satisfying a certain property. Such a problem naturally defines a parameter and therefore the notion of bidimensionality. A minor-bidimensional problem has the *separation property* if it satisfies the following three conditions:

1. If a graph G has k connected components $G_1, G_2, \ldots, G_k$, then an optimal solution for G is the union of optimal solutions for each connected component G_i.

2. There is a polynomial-time algorithm that, given any graph G, given any vertex cut C whose removal disconnects G into connected components $G_1, G_2, \ldots, G_k$, and given an optimal solution S_i to each connected component G_i of $G - C$, computes a solution S for G such that the number of vertices and/or edges in S within the induced subgraph $G[C \cup \cup_{i \in I} V(G_i)]$ consisting of C and some connected components of $G - C$ is $\sum_{i \in I} |S_i| \pm O(|C|)$ for any $I \subseteq \{1, 2, \ldots, k\}$. In particular, the total cost of S is at most $\mathsf{opt}(G - C) + O(|C|)$.

3. Given any graph G, given any vertex cut C, and given an optimal solution opt to G, for any union G' of some subset of connected components of $G - C$, $|\mathsf{opt} \cap G'| = |\mathsf{opt}(G')| \pm O(|C|)$.

For contraction-bidimensional problems, the exact requirements on the problem are slightly different but similarly straightforward. The main distinction is that the connected components are always considered together with the cut C. As a result, the merging algorithm in Condition 2 must take as input a solution to a generalized form of the problem that does not count the cost of including all vertices and edges from the cut C. We refer to [70] for the exact definition of the separation property in this case.

Theorem 1.13 ([70]). *Consider a bidimensional problem satisfying the separation property. Suppose that the problem can be solved on a graph G with n vertices in $f(n, \mathbf{tw}(G))$ time. Suppose also that the problem can be approximated within a factor of α in $g(n)$ time. For contraction-bidimensional problems, suppose further that both of these algorithms also apply to the generalized form of the problem. Then there is a $(1+\varepsilon)$-approximation algorithm whose running time is $O(nf(n, O(\alpha^2/\varepsilon)) + n^3 g(n))$ for the corresponding graph class of the bidimensional problem.*

This result shows a strong connection between subexponential fixed-parameter tractability and approximation algorithms for combinatorial optimization problems on H-minor-free graphs. In particular, this result yields a PTAS for the following minor-bidimensional problems in H-minor-free graphs: feedback vertex set, face cover (defined just for planar graphs), vertex cover, minimum maximal matching, and a series of vertex-removal problems. Furthermore, the result yields a PTAS for the following contraction-bidimensional problems in apex-minor-free graphs: dominating set, edge dominating set, R-dominating set, connected dominating set, connected edge dominating set, connected R-dominating set, and clique-transversal set.

1.11 Half-Integral versus Fractional Multicommodity Flow

Chekuri, Khanna, and Shephard [54] proved that, for planar graphs, the gap between the optimal half-integral multicommodity flow and the optimal fractional multicommodity flow is at most a polylogarithmic factor. Also, they gave a combinatorial proof of the result that, for planar graphs, the gap between the maximum flow and the minimum cut in product multicommodity flow (and thus uniform multicommodity flow) instances is at most a constant factor. The latter result was proved before by Klein, Plotkin, and Rao for H-minor-free graphs using primal-dual methods [120], and has many applications in embeddings of H-minor-free graphs. As mentioned by Chekuri et al. [54], our Theorem 1.6 can be used to generalize the half-integral/fractional gap bound and the combinatorial proof of the max-flow/min-cut gap bound to H-minor-free graphs.

1.12 Thesis Structure

This thesis is organized as follows. We start by demonstrating structural properties of single-crossing-minor-free graphs and their consequences in designing polynomial-time approximation algorithms and subexponential fixed-parameter algorithms for a wide variety of graph problems in Chapters 2 and 3. In Chapter 4, we design fixed-parameter algorithms for the (k, r)-center problem, a generalization of the dominating set problem, on planar graphs and map graphs. At the end of this chapter, we introduce the concept of bidimensionality for planar graphs. In Chapter 5, we formally define the concept of bidimensionality for bounded-genus graphs. In addition, we introduce a general approach for developing algorithms on H-minor-free graphs, based on the corresponding algorithms on bounded-genus graphs and structural results about H-minor-free graphs at the heart of Robertson and Seymour's graph-minors work. Chapters 6 and 7 extend the concept of bidimensionality for apex-minor-free graphs and H-minor-free graphs. In Chapter 8, we prove the linearity of the size of grid minors in terms of the treewidth, by which we improve several combinatorial bounds and running times of algorithms in two previous Chapters 6 and 7. In Chapter 9, we develop the algorithmic theory of vertex separators and its relation to the embeddings of certain metric spaces, by which we improve the approximation factor of treewidth for both general graphs and H-minor-free graphs. We also mention how these improvements can be coupled with bidimensionality to obtain the first polynomial-time approximation schemes for problems like minimum connected dominating set and minimum feedback vertex set in apex-minor-free graphs and H-minor-free graphs. Finally, in Chapter 10, we demonstrate some major directions for future research in the theory of bidimensionality.

It is worth mentioning that the journal versions of Chapters 2, 3, 4, 6, and 7 appeared in order in [72, 74, 64, 65, 62] and the conference versions of Chapters 5, 8, 9, and this chapter appeared in order in [63, 71, 90, 67]. In particular, each chapter is joint work with the set of collaborators listed in the corresponding references.

Chapter 2

Approximation Algorithms for Single-Crossing-Minor-Free Graphs

The development of algorithms for NP-complete problems on restricted classes of graphs has resulted in structural characterizations of algorithmic utility. For example, algorithms for graphs of bounded treewidth rely on techniques using separator properties resulting from tree decompositions. In this chapter we focus on graph classes obtained by excluding a single-crossing graph as a minor. We present a polynomial-time algorithm that determines a *clique-sum decomposition* of such a graph, a representation of the graph as a collection of planar graphs and graphs of small treewidth Our result generalizes previous decomposition results for graphs excluding special single-crossing graphs such as $K_{3,3}$ [20] and K_5 [118]. A second structural property is that of *locally bounded treewidth* which allows a *layer decomposition*, enabling us to represent a graph as a collection of subgraphs, each of bounded treewidth (formal definitions follow in Section 2.3). In order to take advantage of the bound on treewidth of the subgraphs, we give a tree decomposition algorithm for this special case, adding to the toolkit of tree decomposition algorithms for small bounds on treewidth [16, 133, 153] and r-outerplanar graphs [2] that reduce the prohibitively high constant factors found in algorithms for more general graphs [125, 35, 32].

Included among the results in this chapter are several applications of our structural characterizations and algorithms. Using clique-sum decompositions, we obtain the first constant-factor approximation algorithm for treewidth of nonplanar graphs. Furthermore, we use properties of layer decompositions to form polynomial-time approximation schemes for a range of maximization and minimization problems (Section 2.5), as well as fixed-parameter algorithms for dominating set and related problems (Section 3.5).

We present decomposition results followed by algorithms for NP-complete problems. Sections 2.1 through 2.3 present general results from which applications are derived in Sections 2.4 and 2.5. First, in Section 2.1, we introduce the concepts used throughout the chapter (though some of them are defined intuitively in Chapter 1). Next, in Section 2.2, we demonstrate how graphs excluding single-crossing graphs as minors can be characterized using a clique-sum decomposition. The fact that such graphs have bounded local treewidth is established in Section 2.3, which also gives an algorithm for computing the tree decomposition of a local neighborhood in a graph. The rest of the chapter contains results that follow from these properties: an approximation algorithm for treewidth (Section 2.4), polynomial-time approximation schemes for optimization problems (Section 2.5), and summary of our results and directions for further research (Section 2.6). We defer the applications regarding fixed-parameter algorithms for dominating set and its variants to Chapter 3.

2.1 Background

2.1.1 Preliminaries

Recall that a graph is *planar* if it can be drawn in the plane so that its edges intersect only at their endpoints; such a drawing is called an *embedding.* An embedding partitions the plane into connected regions called *faces*; the unbounded region is called the *outer face.* A graph with all vertices on the outer face is called *outerplanar*; a *k-outerplanar graph* has the property that k successive deletions of the vertices on the outer face results in the empty graph.

We consider classes of graphs that are associated with *single-crossing graphs*, as

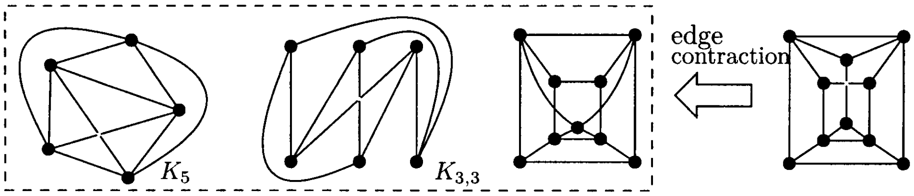

Figure 2-1: Examples of single-crossing graphs

defined by Robertson and Seymour [142]. As defined in Chapter 1, a single-crossing graph is defined as a graph that is a minor of one that can be drawn in the plane with at most one pair of edges crossing. Note that a single-crossing graph may itself not be drawable in this fashion. Figure 2-1 shows to the left three examples of single-crossing graphs; the third one cannot be drawn in the plane with only one crossing, but is obtainable by edge contraction from the graph to the right, which can be so drawn.

The following lemma is a consequence of closure under minors; in contrast, the class of graphs that can be drawn in the plane with at most one pair of edges crossing is not closed under minors, as Figure 2-1 shows.

Lemma 2.1. *If single-crossing-minor-free graph G excludes a single-crossing graph H as a minor, any minor G' of G is also a single-crossing-minor-free graph which excludes H as a minor.*

The following example demonstrates that single-crossing-minor-free graphs can be considerably more complicated than single-crossing graphs. Form a graph on $n = 6k$ vertices by taking k copies of $K_{3,3}$ and adding $k - 1$ edges to connect them into a path-like structure. This graph has $\Theta(n)$ crossings, but is K_5-minor-free. In fact, any graph can be shown to be a single-crossing-minor-free graph, where the excluded single-crossing graph is a sufficiently large grid. However, our algorithms are really only of interest when the excluded graph is small.

2.1.2 Locally Bounded Treewidth

The concept of treewidth can be generalized to that of *locally bounded treewidth* [87], in which each local subgraph has treewidth bounded by a function of r.

Definition 2.2. *The* local treewidth *of a graph G is the function* $\mathbf{ltw}^G : \mathbb{N} \to \mathbb{N}$ *that associates with every $r \in \mathbb{N}$ the maximum treewidth of an r-neighborhood in G. We set* $\mathbf{ltw}^G(r) = \max_{v \in V(G)}\{\mathbf{tw}(G[N_G^r(v)])\}$, *and we say that a graph class $\mathcal{C}$ has* bounded local treewidth (or locally bounded treewidth) *when there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for all $G \in \mathcal{C}$ and $r \in \mathbb{N}$,* $\mathbf{ltw}^G(r) \leq f(r)$. *We say a graph class $\mathcal{C}$ of bounded local treewidth has* linear local treewidth, *if f is linear in r. Finally, for a function $f\colon \mathbb{N} \to \mathbb{N}$, we define the minor-closed class of graphs of bounded local treewidth* $\mathcal{L}(f) = \{G\colon \forall H \preceq G\ \forall r \geq 0,\ \mathbf{ltw}^H(r) \leq f(r)\}$.

Well-known examples of minor-closed classes of graphs of bounded local treewidth are graphs of bounded treewidth, planar graphs, and graphs of bounded genus [87]. Eppstein [87] showed that a minor-closed graph class $\mathcal{C}$ has bounded local treewidth if and only if every graph in $\mathcal{C}$ is H-minor-free for some apex graph H (recall that in any apex graph there exists a vertex whose deletion produces a planar graph). For a simpler proof of this result, the reader might refer to Chapter 6.

The following lemma proves useful in our results:

Lemma 2.3. *For any graph G and subgraph G' of G,* $\mathbf{ltw}^{G'}(k) \leq \mathbf{ltw}^G(k)$, *for any $k \geq 0$.*

Proof. It is enough to observe that for any $v \in G'$ and $k \geq 0$, $N_{G'}^k(v) \subseteq N_G^k(v)$. Thus the removal of vertices of $N_G^k(v) \setminus N_{G'}^k(v)$ from bags of a tree decomposition of $N_G^k(v)$ results in a tree decomposition of $N_{G'}^k(v)$ with width at most that of G. $\square$

2.2 Clique-sum Decompositions

2.2.1 Relating Clique Sums to Treewidth and Local Treewidth

The following lemma shows how the treewidth changes when we apply a clique-sum operation, which will play an important role in our approximation algorithms in Section 2.4.

Lemma 2.4. *For any two graphs G and H,* $\mathbf{tw}(G \oplus H) \leq \max\{\mathbf{tw}(G), \mathbf{tw}(H)\}$.

Proof. We begin by observing that we can form a tree decomposition $TD(G)$ of G of width $\mathbf{tw}(G)$ and a tree decomposition $TD(H)$ of H of width $\mathbf{tw}(H)$. For W the set of vertices of G and H identified during the $\oplus$ operation, W is a clique in G and in H. As vertices in a clique must appear together in a bag in any decomposition of the graph [37], there exist a node α in $TD(G)$ such that $W \subseteq \chi_\alpha$ and a node β in $TD(H)$ such that $W \subseteq \chi_\beta$. Hence, we can form a tree decomposition of width $\max\{\mathbf{tw}(G), \mathbf{tw}(H)\}$ of $G \oplus H$ by adding an edge between α in $TD(G)$ and β in $TD(H)$. □

To extend the result to graphs of bounded local treewidth in Lemma 2.9 (Section 2.3), in Lemma 2.5 we establish treewidth properties for neighborhoods of graphs.

Lemma 2.5. *For any graph G, any clique R of G, any $v \in R$, and any $k \geq 0$,* $\mathbf{tw}(G[N_G^k(R)]) \leq \mathbf{tw}(G[N_G^{k+1}(v)])$.

Proof. We note that all vertices in $R-v$ are at distance 1 from v. Therefore $N_{G_1}^k(R) \subseteq N_{G_1}^{k+1}(v)$, and the result follows from Lemma 2.3. □

2.2.2 Decomposition Algorithm

The main theorem of this section is a constructive version of the following non-algorithmic result of Robertson and Seymour, itself used in our algorithm.

Theorem 2.6. [142] *For any single-crossing graph H, there is an integer $c_H \geq 4$ (depending only on H) such that every H-minor-free graph can be obtained by* 0-, 1-, 2- *or* 3-*sums of planar graphs and graphs of treewidth at most c_H.*

In the remainder of the chapter we will assume that c_H is the smallest integer for which Theorem 1 holds. Although previous algorithms have been developed for decomposing specific single-crossing-minor-free graphs into series of clique sums (Asano [20] gave an $O(n)$-time construction for $K_{3,3}$-minor-free graphs and Kézdy and McGuinness [118] gave an $O(n^2)$-time construction for K_5-minor-free graphs), ours is the first general algorithm. For more examples of graph classes that can be characterized by clique-sum decompositions, see the work of Diestel [77, 78]. Theorem 2.8 below describes a constructive algorithm to obtain a clique-sum decomposition of a single-crossing-minor-free graph G that satisfies the additional property that the smaller graphs are minors of G. The additional property is crucial for designing approximation algorithms in Section 2.4.

To form a clique-sum decomposition of graphs that are minors of the original graph, we consider graphs formed by first removing a set of vertices from the graph and then reinserting the removed vertices in each resulting connected component. More formally, we define a subset $S \subseteq V$ to be a *k-cut* if the induced subgraph $G[V-S]$ is disconnected and $|S| = k$, and to be a *strong k-cut* if in addition $G[V-S]$ either has more than two connected components or each component has more than one vertex. For S a strong cut that separates G into components $G_1, \ldots, G_h$, we form the *augmented components induced by S*, denoted $G_i \cup K(S)$ for $1 \leq i \leq h$, as the graphs obtained from graphs $G[V(G_i) \cup S]$ by adding an edge between each pair of nonadjacent vertices in S. Each augmented component will contain as a subgraph a clique on the vertices in S. The influence of strong cuts on augmented components is fundamental in the proof of our theorem. By introducing a less strict definition of strong cuts, we obtain a stronger version of an earlier lemma [118].

Lemma 2.7. *Let S be a strong 3-cut of a 3-connected graph $G = (V, E)$, and let $G_1, G_2, \ldots, G_h$ denote the h components of $G[V-S]$. Then each augmented component of G induced by S, $G_i \cup K(S)$, is a minor of G.*

Proof. We first consider the case in which $h \geq 3$. By symmetry, it will suffice to show that $G_1 \cup K(S)$ is a minor of G by forming the graph by a series of contractions in G. Starting with G, we first contract all edges of G_2 through G_h to obtain super-vertices y_2 through y_h (we use the term *super-vertices* to denote vertices that are the result of the contraction of all the edges in specific connected subgraphs of G). Because G is 3-connected, each super-vertex is adjacent to all vertices x_1, x_2, and x_3 in S. We now contract the edge $\{x_2, y_2\}$ to form a super-vertex x'_2 and contract edges between y_3 through y_h and x_3 to form a super-vertex x'_3. Again since each y_i was adjacent to each x_j, we can conclude that x_1, x'_2, and x'_3 form a clique and hence the resulting graph is the augmented component $G_1 \cup K(S)$, as needed.

If instead $h = 2$, then by the definition of a strong cut G_1 and G_2 each contain at least two vertices. As in the previous case, it will suffice to show that $G_1 \cup K(S)$ is a minor of G, as the argument for $G_2 \cup K(S)$ is symmetric. Again, we demonstrate a series of contractions that results in G_1 as well as a clique on the vertices x_1, x_2, and x_3 of S. We consider separately the cases in which G_2 is a tree and G_2 contains a cycle.

If G_2 is a tree, then because G is 3-connected, there is a vertex y_1 in G_2 that neighbors x_1, and similarly a vertex $y_3 \neq y_1$ in G_2 that neighbors x_3. We contract edges in G_2 until there remain two super-vertices y'_1 and y'_3, connected to x_1 and x_3 respectively. Since G_2 is connected but acyclic, there will be a single edge between y'_1 and y'_3. Moreover, since G is 3-connected, there is an edge between x_2 and either y'_1 or y'_3, say y'_1. Finally, again because G is 3-connected, y'_3 must be adjacent to a vertex of S other than x_3, that is, either x_2 or x_1. In either case we can form a clique on the vertices in S, by contracting edges $\{x_1, y'_1\}$ and $\{x_3, y'_3\}$ if y'_3 is adjacent to x_2 or by contracting the edges $\{x_2, y'_1\}$ and $\{x_3, y'_3\}$ if y'_3 is adjacent to x_1.

Finally suppose that G_2 has a cycle C. We claim that there are three vertex-disjoint paths connecting three vertices of C to three vertices of S in G_2. By contracting these paths and then contracting edges of C to form a triangle, we have a clique on the vertices of S as desired. To prove the claim, we augment the graph G by adding a vertex v_1 connected to every vertex in S, and by adding a vertex v_2 connected to every vertex in C. Because $|S| = 3$ and $|C| \geq 3$, the augmented graph is still 3-connected. Therefore there exist at least three vertex-disjoint paths from v_1 to v_2. Each internal vertex on each of these paths must be in G_2, and each path must contain an edge from v_1 to a vertex of S and an edge from a vertex of C to v_2. We obtain the desired paths by removing v_1 and v_2 from each path. $\square$

Theorem 2.8. *For any graph G excluding a single-crossing graph H as a minor, we can construct in $O(n^4)$ time a series of clique-sum operations $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is a minor of G and is either a planar graph or a graph of treewidth at most c_H. Here each $\oplus$ is a 0-, 1-, 2- or 3-sum.*

Proof. The algorithm proceeds by recursively determining connectivity of subgraphs of the original graph, where different decompositions are used depending on the type of cut. When considering graph G, we first determine whether it is 1-, 2-, or 3-connected. If G is disconnected, each of its connected components is considered separately, and all are joined by 0-sums to form G. If G has a 1-cut, 2-cut, or strong 3-cut S, we recursively apply the algorithm on the augmented components induced by S, applying, respectively, 1-, 2-, or 3-sums. Finally, if G is 3-connected but has no strong 3-cut, then we claim that it is either planar or has treewidth at most c_H.

We first prove the correctness of the outline above, and later fill in the algorithmic details and analyze the running time. To show that the recursive application of the algorithm will yield a correct solution, we need to show that each subgraph created is a minor of G (and hence is H-minor-free by Lemma 2.1). A connected component of a disconnected graph or an augmented component resulting from a 1-cut is a subgraph of G, and hence a minor. For an augmented component formed by a 2-cut, the 2-connectivity of G guarantees that any component must connect to both vertices in the cut, and hence edges can be contracted to yield the edge added in the augmentation. Lemma 2.7 handles the case in which there is a strong 3-cut.

Finally, we need to prove that if the graph G is 3-connected but has no strong 3-cut, then either the treewidth of G is at most c_H or G is planar. Suppose instead that neither of these properties hold. Since G is H-minor-free and 3-connected, by Theorem 2.6, G can be obtained by 3-sums of a sequence of graphs $\mathcal{C}$, where each graph in $\mathcal{C}$ is either planar or of treewidth at most $c_H \geq 4$. As G has no strong 3-cut, for any join set S in the clique-sum decomposition $G - S$ may contain at most one component with more than one vertex, and hence at most one graph in $\mathcal{C}$ can have more than four vertices. If every graph in $\mathcal{C}$ has at most four vertices and hence treewidth at most 3 (and hence less than c_H), then by Lemma 2.4, G would have treewidth less than c_H, contradicting our assumption. We can thus conclude that $\mathcal{C}$ contains subgraphs of K_4 and one planar graph of at least five vertices and of treewidth greater than c_H.

To complete the proof of correctness, we will show that our assumption about $\mathcal{C}$

yields a contradiction, that is, that G has a strong 3-cut. Since G is not planar but every graph in $\mathcal{C}$ is planar, during the clique-sum operations forming G, there exists a graph $J \in \mathcal{C}$ and a 3-sum $G'' = G' \oplus J$ with join set S such that G' is planar but G'' is not planar. It is not possible to form planar embeddings of both J and G' such that S forms the outer face, since if this were possible, the two embeddings could be joined to form a planar embedding of G'' (e.g. J would be embedded inside the triangle and G' outside). We can thus conclude that there are at least three components in $G'' - S$, which implies that S is a strong 3-cut in G'' and hence in G, a contradiction.

To analyze the running time of the algorithm, we first recall that any graph G excluding an r-clique as a minor cannot have more than $(0.319+o(1))(r\sqrt{\log r})|V(G)|$ edges [161]. This implies that for any single-crossing-minor-free graph G, $|E(G)| = O(|V(G)|)$.

To run the algorithm, we apply algorithms to obtain all connected components and 1-cuts in linear time [158], all 2-cuts [114, 134], and all $O(n^2)$ 3-cuts in $O(n^2)$ time [115]. Checking whether a particular 3-cut is strong can be accomplished in $O(n)$ time using depth-first search. All other operations, including checking if a graph is planar or has treewidth at most c_H, can be performed in linear time [166, 32].

To set up recurrence relations, we make the assumption that for each cut we split a graph into two 0-, 1-, or 2-connected components or into at most three 3-connected components at a particular iteration. The running time of one iteration, excluding recursive calls, is $O(n)$. For $T(n)$ the running time on an input of size n, for a 0-sum involving a component of size n_1 we obtain the equation:

$$T(n) = T(n_1) + T(n - n_1) + O(n), \quad n_1 \geq 2.$$

The recursive calls for a 1-cut yield the following equation

$$T(n) = T(n_1) + T(n - n_1 + 1) + O(n), \quad n_1 \geq 2$$

where n_1 and $n - n_1 + 1$ are the sizes of the two augmented components. Similarly, for recursive calls for a 2-cut, we have

$$T(n) = T(n_1) + T(n - n_1 + 2) + O(n), \quad n_1 \geq 3.$$

For recursive calls for a strong 3-cut with exactly two components, we have

$$T(n) = T(n_1) + T(n - n_1 + 3) + O(n^3), \quad n_1 \geq 4.$$

Finally, if we have recursive calls for a strong 3-cut with at least three components, we have

$$T(n) = T(n_1)+T(n_2)+T(n-n_1-n_2+6)+O(n^3), \quad 4 \leq n_1, n_2, n-n_1-n_2+6 \leq n-2$$

where n_1, n_2, and $n - n_1 - n_2 + 6$ are the sizes of the augmented components (the last being the third and all subsequent components taken together). The additive terms (+1, +2, +3, +6) are due to the duplication of the vertices of the cut in the augmented components. Solving this recurrence gives a worst-case running time of $O(n^4)$. □

Even for excluded graphs H where c_H is huge, the value of c_H does not contribute to the asymptotic complexity of the algorithm presented above. However, it does contribute heavily to the constant hidden in the order notation. In some contexts, it may make sense to replace Bodlaender's linear-time algorithm for determining treewidth exactly with an approximation. Amir [11] gives an algorithm running in time $O(2^{4.38c_H} n^2 c_H)$ which either returns a tree-decomposition of width at most $4c_H$ or answers that the treewidth is more than c_H. This can be used to prove a version of Theorem 2 with c_H replaced by $4c_H$ but whose dependence on c_H is more reasonable. Similar substitutions will be possible in our approximation algorithms, discussed in Section 2.5.

2.3 Locally Bounded Treewidth of Single-Crossing-Minor-Free Graphs

In this section we establish the locally bounded treewidth of single-crossing-minor-free graphs, which provides the structure on which the approximation schemes of Section 2.5 are built. First we demonstrate how clique sums and local treewidth are correlated. Next, we discuss layer decompositions and present a tree decomposition algorithm for a subgraph induced by a sequence of consecutive layers.

2.3.1 Bounded Local Treewidth

Lemma 2.9. *If G_1 and G_2 are graphs such that $\mathbf{ltw}^{G_1}(r) \leq f(r)$ and $\mathbf{ltw}^{G_2}(r) \leq f(r)$ for a function $f(r) \geq 0$ for all $r \in \mathbb{N}$, and $G = G_1 \oplus_k G_2$, then $\mathbf{ltw}^{G}(r) \leq f(r)$.*

Proof. To show $\mathbf{ltw}^{G}(r) \leq f(r)$, we prove that for any $v \in V(G)$ and for all $r \geq 0$, $\mathbf{tw}(G[N_G^r(v)]) \leq f(r)$. We use W to denote the join set of $G_1 \oplus_k G_2$ and without loss of generality, we assume v is from G_1. As the claim holds trivially for $r = 0$, in the remainder of the proof we assume $r > 0$. Moreover, since if $N_G^r(v)$ contains only vertices originally from G_1, the result follows from the fact that $\mathbf{ltw}^{G_1}(r) \leq f(r)$, we assume that $N_G^r(v)$ contains vertices from G_2 that are not in W. We consider two cases, depending on whether $v \in W$.

If $v \in W$, then $N_G^r(v) \subseteq N_{G_1}^r(v) \cup N_{G_2}^r(v)$. In addition, since $r \geq 1$ and vertices of W form a clique in G_i for $i = 1, 2$, $W \subseteq N_{G_i}^r(v)$. Using these two facts, we conclude that $G[N_G^r(v)]$ is a subgraph of $G_1[N_{G_1}^r(v)] \oplus G_2[N_{G_2}^r(v)]$ over the join set W. Thus, by Lemmas 2.3 and 2.4, we know

$$\mathbf{tw}(G[N_G^r(v)]) \leq \max\{\mathbf{tw}(G_1[N_{G_1}^r(v)]), \mathbf{tw}(G_2[N_{G_2}^r(v)])\} \leq f(r).$$

We now consider the case in which $v \notin W$. Each vertex in $N_G^r(v)$ is either in G_1, and hence is in $N_{G_1}^r(v)$, or it is in G_2, and hence is in $N_G^r(v) - W$. To further describe the latter set, we consider the distance between v and any vertex u in the set. Since W is the set of vertices shared by G_1 and G_2 in G, at least one vertex of W is on the shortest path from v to u in G and hence is at distance of at most $r-1$ from v, hence $W \cap N_{G_1}^{r-1}(v) \neq \emptyset$. We let p be the minimum distance between v and any vertex in the set $W \cap N_{G_1}^{r-1}(v)$ and observe that since v in not in W, we can conclude that $1 \leq p \leq r-1$. We further observe that since each vertex u in $N_G^r(v) - W$ is at distance at most r from v, it must be within distance $r-p$ of some vertex of W, or $u \in N_{G_2}^{r-p}(W)$. Thus $N_G^r(v) \subseteq N_{G_1}^r(v) \cup N_{G_2}^{r-p}(W)$.

To complete the proof, we use the characterization of $N_G^r(v)$ as a subset of $N_{G_1}^r(v) \cup N_{G_2}^{r-p}(W)$ to obtain an upper bound on its treewidth. First we show that $G_1[N_{G_1}^r(v)] \oplus G_2[N_{G_2}^{r-p}(W)]$ can be formed using the join set W, as W is a subset of each of the constituent graphs (each vertex of W is at distance at most r from v in G_1 since at least one vertex of W is at distance $p \leq r-1$ from v and vertices of W form a clique in G_1). Since $N_G^r(v) \subseteq N_{G_1}^r(v) \cup N_{G_2}^{r-p}(W)$, as shown in the previous paragraph, $G[N_G^r(v)]$ is a subgraph of $G_1[N_{G_1}^r(v)] \oplus G_2[N_{G_2}^{r-p}(W)]$, and hence we can apply Lemma 2.4 to obtain the following result:

$$\mathbf{tw}(G[N_G^r(v)]) \leq \max\{\mathbf{tw}(G_1[N_{G_1}^r(v)]), \mathbf{tw}(G_2[N_{G_2}^{r-p}(W)])\}. \tag{2.1}$$

By Lemma 2.3, since $p \geq 1$ clearly $G_2[N_{G_2}^{r-p}(W)]$ is a subgraph of $G_2[N_{G_2}^{r-1}(W)]$, and hence

$$\mathbf{tw}(G_2[N_{G_2}^{r-p}(W)]) \leq \mathbf{tw}(G_2[N_{G_2}^{r-1}(W)]). \tag{2.2}$$

Combining (2.1), (2.2), and the fact that $\mathbf{tw}(G_1[N_{G_1}^r(v)]) \leq f(r)$ (our assumption about G_1), we obtain

$$\mathbf{tw}(G[N_G^r(v)]) \leq \max\{f(r), \mathbf{tw}(G_2[N_{G_2}^{r-1}(W)])\}. \tag{2.3}$$

Since W is a clique in G_2, by Lemma 2.5,

$$\mathbf{tw}(G_2[N_{G_2}^{r-1}(W)]) \leq \mathbf{tw}(G_2[N_{G_2}^r(v)]) \leq f(r). \tag{2.4}$$

Finally, as a consequence of (2.3) and (2.4), we conclude that $\mathbf{tw}(G[N_G^r(v)]) \leq f(r)$,

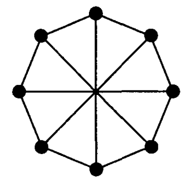

Figure 2-2: The graph V_8.

as needed to complete the proof of the lemma. $\square$

Theorem 2.10 demonstrates our main result on the local treewidth of single-crossing-minor-free graphs.

Theorem 2.10. *For any single-crossing-minor-free graph G excluding a single-crossing graph H as a minor and for all $r \geq 0$,* $\mathbf{ltw}^G(r) \leq 3r + c_H$.

Proof. By Theorem 2.6, we can assume $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is either a planar graph or a graph of treewidth at most c_H; we prove the theorem by induction on m. In the base case, if G_1 is a planar graph then $\mathbf{ltw}^G(r) = \mathbf{ltw}^{G_1}(r) = 3r - 1 \leq 3r + c_H$, $c_H \geq 0$, as the treewidth of a r-outerplanar graph is at most $3r - 1$ [2]. If instead G_1 has treewidth at most c_H, then $\mathbf{ltw}^G(r) = \mathbf{ltw}^{G_1}(r) = c_H \leq 3r + c_H$, $r \geq 0$. To prove the general case, we assume the induction hypothesis is true for $m = h$, and we prove the hypothesis for $m = h + 1$ by setting $G' = G_1 \oplus G_2 \oplus \cdots \oplus G_h$ and $G'' = G_{h+1}$. By the induction hypothesis, $\mathbf{ltw}^{G'}(r) \leq 3r + c_H$ and $\mathbf{ltw}^{G''}(r) \leq 3r + c_H$; by applying Lemma 2.9, we conclude that $\mathbf{ltw}^G = \mathbf{ltw}^{G' \oplus G''}(r) \leq 3r + c_H$, as needed. $\square$

Using the fact that K_5 and $K_{3,3}$ are single-crossing graphs (Figure 2-1), we observe that K_5-minor-free graphs and $K_{3,3}$-minor-free graphs are single-crossing-minor-free graphs. Although generalized by Theorem 2.8 for single-crossing-minor-free graphs, for more precise results we rely on Wagner's characterizations [164]. He proved that a graph is $K_{3,3}$-minor-free if and only if it can be obtained from planar graphs and K_5 by 0-, 1-, and 2-sums and that a graph is K_5-minor-free if and only if it can be obtained from planar graphs and V_8 (the graph obtained from a cycle of length 8 by joining each pair of diagonally opposite vertices by an edge, shown in Figure 2-2) by 0-, 1-, 2-, and 3-sums. Since both K_5 and V_8 have treewidth four, the value of constant c_H in the proof of Theorem 2.10 is four, and we have:

Corollary 2.11. *If G is a K_5-minor-free or $K_{3,3}$-minor-free graph then* $\mathbf{ltw}^G(r) \leq 3r + 4$.

2.3.2 Local Treewidth and Layer Decompositions

To take advantage of the bound on local treewidth, we define a layer decomposition, prove a bound on the treewidth of a subgraph induced on consecutive layers, and then provide an algorithm that forms a tree decomposition of such a subgraph. The concept of the kth outer face in planar graphs can be replaced by the concept of the kth layer (or level) in graphs of locally bounded treewidth. The kth layer (L_k) of a graph G consists of all vertices at distance k from an arbitrary fixed vertex $\hat{v}$ of $V(G)$. We denote *consecutive layers from i to j* by $L[i,j] = \bigcup_{i \leq k \leq j} L_k$, and call such a representation a *layer decomposition.*

Theorem 2.12. *For any graph G excluding a single-crossing graph H as a minor, the treewidth of $G[L[i,j]]$ is bounded above by $3(j-i+1)+c_H$.*

Proof. By contracting the connected subgraph $G[L[0,i-1]]$ to a vertex v' and applying Lemma 2.1, we obtain another H-minor-free graph G'. As all vertices at distance d, $i \leq d \leq j$, from $\hat{v}$ in G are at distance d', $1 \leq d' \leq j-i+1$, from v' in G' and all vertices at distance more than j from $\hat{v}$ in G are at distance more than $j-i+1$ from v' in G', we have $G[L[i,j]] = G'[L[1,j-i+1]]$. Thus $\mathbf{tw}(G[L[i,j]]) = \mathbf{tw}(G'[L[1,j-i+1]])$. Since all vertices of $L[1,j-i+1]$ in G' are in the $j-i+1$-neighborhood of v', $\mathbf{tw}(G'[L[1,j-i+1]]) \leq \mathbf{tw}(G'[N_{G'}^{j-i+1}(v')])$. By the definition of local treewidth, $\mathbf{tw}(G'[N_{G'}^{j-i+1}(v')]) \leq \mathbf{ltw}^{G'}(j-i+1)$. Finally by Theorem 2.10, we have $\mathbf{ltw}^{G'}(j-i+1) \leq 3(j-i+1)+c_H$. Using these facts, $\mathbf{tw}(G[L[i,j]]) \leq 3(j-i+1)+c_H$, as desired. □

Theorem 2.12 gives an upper bound on the treewidth of consecutive layers from i to j, but does not provide a constructive algorithm to obtain a tree decomposition of this width.

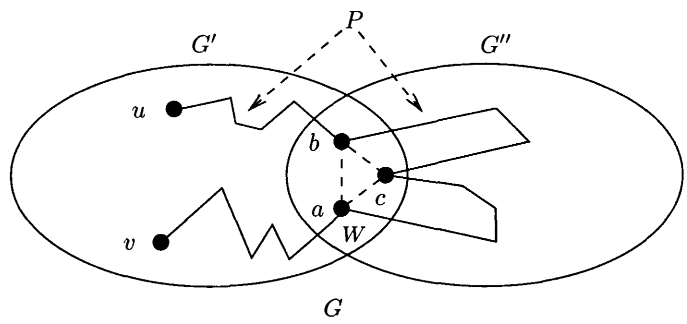

Figure 2-3: The replacement of the part of path P between a and b by edge $\{a,b\}$.

Although we can construct a tree decomposition of width $3(j-i)+c_H$ in linear time using Bodlaender's algorithm [32], again the hidden constant factor will depend on this entire width. Below we show how to reduce the constant to depend only on c_H, permitting substitution of approximation algorithms as was done at the end of Section 2.2.

Before stating the main theorem on construction of a tree decomposition of consecutive layers, we present a simple lemma.

Lemma 2.13. *For $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$, if there exists a vertex $v \in V(G)$ such that each vertex of G is at distance at most r from v, then in each G_i, $1 \le i \le m$, there exists a vertex v_i such that each vertex of G_i is at distance at most r from v_i.*

Proof. We use induction on m, the number of G_i's. If $m = 1$, the basis of induction is clearly true. We assume the induction hypothesis is true for $m \le h$, and we prove the hypothesis for $m = h+1$. We suppose $G = G' \oplus G''$, with join set W, where $G' = G_1 \oplus G_2 \oplus \cdots \oplus G_h$ and $G'' = G_{h+1}$. In order to apply the induction hypothesis to G' and G'', we will need to find vertices v' and v'' in G' and G'' such that each vertex of G' is at distance at most r from v' and each vertex of G'' is at distance at most r from v''.

In order to apply the induction hypothesis to G' and G'', it will suffice to show that there is a path of length at most r from v to each vertex u in G' such that each vertex of the path is in G', that is $v = v'$ as defined in the previous paragraph (an analogous argument can be used to show the existence of a path in G'' to any vertex in G'').

Suppose instead that every path of length at most r from v to u passed through at least one vertex w in $V(G'') - V(G')$, and consider one such path P. Clearly v and u are not both in W, since otherwise there would exist an edge $\{v, u\}$ in the clique with vertex set W. Without loss of generality we assume $u \in V(G') - V(G'')$, and observe that since the path from v to w and the path from w to u must pass through W, we can define a and b to be the first and last vertices in W on the path in order from v to u (see Figure 2-3), where possibly $v = a$ or $a = b$ (or both). We can then form a new path P' consisting of the subpath from v to a, the edge $\{a, b\}$ if $a \neq b$ (which must exist since the vertices of W form a clique in G'), and the subpath from b to u. The path P' has length at most the length of P and is entirely in G', contradicting our assumption and completing the proof. $\square$

We are ready to present our algorithm for construction of a tree decomposition for a constant number of consecutive layers.

Theorem 2.14. *For any single-crossing-minor-free graph G, we construct a tree decomposition for $G[L[i,j]]$ of treewidth $3(j-i+1)+c_H$ in $O((j-i+1)^3 \cdot n + n^4)$ time; for a $K_{3,3}$-minor-free or K_5-minor-free graph G, the running time can be reduced to $O((j-i+1)^3 \cdot n)$ or $O((j-i+1)^3 \cdot n + n^2)$, respectively.*

Proof. As in the proof of Theorem 2.12, we contract the connected subgraph $G[L[0, i-1]]$ to a vertex v' and obtain another single-crossing-minor-free graph G' such that $G[L[i,j]] = G'[L[1, j-i+1]]$. By Lemma 2.1, the graph $G'' = G'[L[0, j-i+1]]$ is a single-crossing-minor-free graph excluding the same H and by the definition of layers each vertex in G'' is at distance at most $j-i+1$ from v'. By Theorem 2.8, we can determine a set of clique-sum operations of graph G'' in $O(n^4)$ time (improved to $O(n)$ for G $K_{3,3}$-minor-free using the result of Asano [20] and $O(n^2)$ for G K_5-minor-free using the result of Kézdy and McGuinness [118]).

After determining a set of clique-sum operations of $G'' = G_1 \oplus G_2 \oplus \cdots \oplus G_m$, we construct a tree decomposition for each G_i, $1 \leq i \leq m$. If G_i is a graph of treewidth at most c_H, we can easily construct a tree decomposition of constant width in linear time [32] (for the special cases, K_5 or V_8, a constant time construction is possible).

We now consider the case in which G_i is a planar graph. By Lemma 2.13, in each G_i, there exists a vertex v_i such that each vertex in G_i is at distance at most $j-i+1$ from v_i. It is known that if a planar graph has a rooted spanning tree T in which the longest path has length d, then a tree decomposition of the graph with width at most $3d$ can be found in time $O(dn)$ [23, 86]. Since each vertex in G_i is at distance at most $j-i+1$ from v_i, by breadth-first search we can construct a spanning tree rooted at v_i with the longest path of length at most $j-i+1$. Hence we can construct a tree decomposition for G_i of treewidth $3(j-i+1)$ in time $O((j-i+1) \cdot |V(G_i)|)$.

Having tree decompositions of G_i's, $1 \leq i \leq m$, in the rest of the algorithm, we glue together the tree decompositions of G_i's using the construction given in the proof of Lemma 2.4. To this end, we introduce an array *Nodes* indexed by all subsets of $V(G'')$ of size at most three. In this array, for each subset whose elements form a clique, we specify a node of the tree decomposition which contains this subset. We note that for each clique C in G_i, there exists a node z of $TD(G'')$ such that all vertices of C appear in the bag of z [37]. This array is initialized as part of a preprocessing stage of the algorithm. Now, for the $\oplus$ operation between $G_1 \oplus \cdots \oplus G_h$ and G_{h+1} over the join set W, using array *Nodes*, we find a node α in the tree decomposition of $G_1 \oplus \cdots \oplus G_h$ whose bag contains W. Since we have the tree decomposition of G_{h+1}, and W is a clique of size at most three that must appear in some bag in any tree decomposition, we can find the node α' of the tree decomposition whose bag contains W by brute force search over all subsets of bags of size at most three. Simultaneously, we update array *Nodes* by subsets of $V(G'')$ which form a clique and appear in bags of the tree decomposition of G_{h+1}. Then we add an edge between α and α'. As the number of nodes in a tree decomposition of G_{h+1} is $O(|V(G_{h+1})|)$ and each bag has size at most $3(j-i+1)$ (and thus there are at most $O((j-i+1)^3)$ choices for a subset of size at most three), this operation takes $O((j-i+1)^3 \cdot |V(G_{h+1})|)$ time for G_{h+1}.

The claimed running time follows from the time required to determine a set of clique-sum operations, the time required to construct tree decompositions, the time needed to glue tree decompositions together and the fact that $\sum_{i=1}^{m} |V(G_i)| =$

$O(|V(G'')|)$. Here we note that the only difference between the running times for the general algorithm and those for $K_{3,3}$-minor-free or K_5-minor-free graphs is the time required to determine a set of clique-sum operations ($O(n)$ time for the former graphs and $O(n^2)$ time for the latter graphs). The rest of the algorithm requires linear time for all single-crossing-minor-free graphs.

Finally, we prove that the width of the constructed tree decomposition of G'' is $3(j - i + 1) + c_H$. We use induction on m, the number of G_i's, where $G'' = G_1 \oplus G_2 \oplus \cdots \oplus G_m$. For $m = 1$, G_1 is either a planar graph of treewidth at most $3(j-i+1)$ or a graph of treewidth at most c_H. In both cases the basis of the induction is true. We assume the induction hypothesis is true for $m = h$, and we prove the hypothesis for $m = h + 1$. For $\tilde{G} = G_1 \oplus G_2 \oplus \cdots \oplus G_h$, $G'' = \tilde{G} \oplus G_{h+1}$. By the induction hypothesis, $\tilde{G}$ and G_{h+1} each have treewidth at most $3(j - i + 1) + c_H$. We can then apply Lemma 2.4 to conclude that the treewidth of G'' is also at most $3(j - i + 1) + c_H$, as needed to complete the proof. □

2.4 Approximating Treewidth

A large amount of effort has been put into determining treewidth, which is NP-complete even if we restrict the input graph to graphs of bounded degree [38], cocomparability graphs [14, 108], bipartite graphs [121], or the complements of bipartite graphs [14]. However, treewidth can be computed exactly in polynomial time for chordal graphs, permutation graphs [36], circular-arc graphs [156], circle graphs [121], distance-hereditary graphs [44], and for graphs of a fixed treewidth [32].

A constant factor approximation algorithm for treewidth of planar graphs is known. This approximation algorithm is a consequence of the polynomial-time algorithm given by Seymour and Thomas [155] for computing the parameter *branchwidth*, whose value approximates treewidth within a factor of 1.5. Using the notions of branchwidth and clique-sum decomposition, we demonstrate a polynomial-time algorithm that approximates within a constant factor the treewidth of any single-crossing-minor-free graph. Using a different approach, this result will be generalized

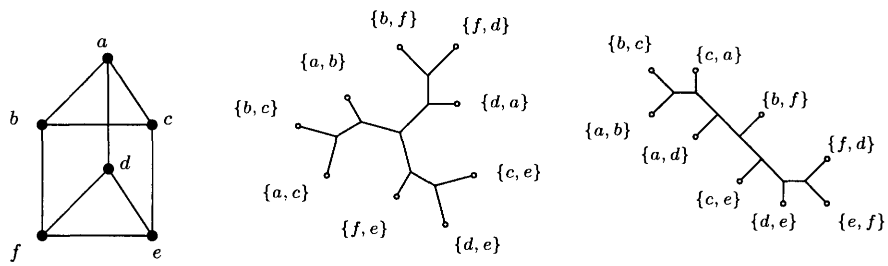

Figure 2-4: A graph and two branch decompositions of it. The first has width 4 and the second has width 3.

for all H-minor-free graphs, for a fixed H, in Chapter 9.

Analogous to the relationship between treewidth and tree decompositions, the notion of branchwidth is related to a decomposition based on the edges. A *branch decomposition* of a graph G is a pair (T, τ), where T is a tree with vertices of degree 1 or 3 and τ is a bijection from the set of leaves of T to $E(G)$. The *order* of an edge e in T is the number of vertices $v \in V(G)$ such that there are leaves t_1, t_2 in T in different components of $T(V(T), E(T) - e)$ with $\tau(t_1)$ and $\tau(t_2)$ both containing v as an endpoint. The *width* of (T, τ) is the maximum order over all edges of T, and the *branchwidth* of G, $\mathbf{bw}(G)$, is the minimum width over all branch decompositions of G (if $|E(G)| \leq 1$, we define the branchwidth to be 0; if $|E(G)| = 0$, then G has no branch decomposition; if $|E(G)| = 1$, then G has a branch decomposition consisting of a tree with one vertex and the width of this branch decomposition is considered to be 0). It is well-known that, if $H \preceq G$ or $H \preceq_c G$, then $\mathbf{bw}(H) \leq \mathbf{bw}(G)$. Figure 2-4 provides examples of branch decompositions.

The following result of Robertson and Seymour [145] relates branchwidth to treewidth.

Theorem 2.15 ([145], Section 5). *For any connected graph G where $|E(G)| \geq 3$, $\mathbf{bw}(G) \leq \mathbf{tw}(G) + 1 \leq \frac{3}{2}\,\mathbf{bw}(G)$.*

We make use of an approximation algorithm for computing treewidth of planar graphs as one of two "base cases" in our algorithm for single-crossing-minor-free graphs. While it remains an open question whether there exists a polynomial-time constant-factor approximation algorithm for computing the treewidth of general graphs, the branchwidth of a planar graph can be computed in polynomial time.

Theorem 2.16. ([155], Sections 7 and 9) *One can construct an algorithm that, given a planar graph G,*

1. *computes in $O(n^2 \log n)$ time the branchwidth of G; and*

2. *computes in $O(n^4)$ time a branch decomposition of G with optimal width.*

Theorem 2.17. *One can construct an algorithm that, given a planar graph G,*

1. *computes in $O(n^2 \log n)$ time a value k with $tw(G) \leq k \leq \frac{3}{2}\,\mathbf{tw}(G)$; and*

2. *computes in $O(n^4)$ time a tree decomposition D of width k of G, where $\mathbf{tw}(G) \leq k \leq \frac{3}{2}\,\mathbf{tw}(G) + 1$.*

Proof. The proof is straightforward using $O(|E(G)|^2)$ algorithms of Robertson and Seymour [145] which convert a branch decomposition of width at most b to a tree decomposition of width at most $\frac{3}{2}b - 1$, and convert a tree decomposition of width at most k to a branch decomposition of width at most $k + 1$. □

The main theorem of this section relies on Theorem 2.17 as well as clique-sum decompositions.

Theorem 2.18. *For any single-crossing graph H, we can construct an algorithm that, given an H-minor-free graph as input, outputs in $O(n^4)$ time a tree decomposition of G of width k where $\mathbf{tw}(G) \leq k \leq \frac{3}{2}\,\mathbf{tw}(G) + 1$.*

Proof. The algorithm consists of the following four steps:

Step 1: Let G be a graph excluding a single-crossing graph H. By Theorem 2.8, we can obtain a clique-sum decomposition $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is a minor of G and is either a planar graph or a graph of treewidth at most c_H. According to the same theorem, this step can be executed in $O(n^4)$ time. Let B be the set of the indices of the bounded treewidth components and P be the set of planar components: $B = \{i \mid 1 \leq i \leq m,\ \mathbf{tw}(G_i) \leq c_H\}$, $P = \{1, \ldots, m\} - B$.

Step 2: By Theorem 2.17, we can construct, for any $i \in P$, a tree decomposition $TD(G_i)$ of G_i with width k_i and such that

$$\mathbf{tw}(G_i) \ \leq k_i \leq \ \tfrac{3}{2}\,\mathbf{tw}(G_i) + 1 \quad \text{for all } i \in P. \tag{2.5}$$

The construction of each of these tree decompositions takes $O(|V(G_i)|^4)$ time. Because $m \leq n$ and $\sum_{1\leq i\leq m} |V(G_i)| = O(n)$, (it is simple to prove this by induction looking at the sizes of the components in the recurrences used in the proof of Theorem 2.8) the total time for this step is $O(n^4)$.

Step 3: Using Bodlaender's algorithm [32], for any $i \in B$, we can obtain a tree decomposition of G_i with minimum width k_i in linear time, where the hidden constant depends only on c_H. Combining (2.5) with the fact that $\mathbf{tw}(G_i) = k_i$ for each $i \in B$, we obtain

$$\mathbf{tw}(G_i) \ \leq k_i \leq \ \tfrac{3}{2}\,\mathbf{tw}(G_i) + 1 \quad \text{for all } i \in \{1, \ldots, m\}. \tag{2.6}$$

Step 4: Now that we have tree decompositions $TD(G_i)$ of each G_i, we glue them together using the construction given in the proof of Lemma 2.4, as detailed in Theorem 2.14. In this way, we obtain a tree decomposition of G that has size $k = \max\{k_i \mid 1 \leq i \leq m\}$. Combining this equality with (2.6), we have

$$\max\{\mathbf{tw}(G_i) \ \mid i = 1, \ldots, m\} \ \leq k \leq \tfrac{3}{2} \max\{\mathbf{tw}(G_i) \mid i = 1, \ldots, m\} + 1 \tag{2.7}$$

To see that this step can be executed in $O(n^4)$ time, we observe that as described in Theorem 2.14, for each of the $O(|V(G)|)$ nodes in the tree decomposition, we execute a brute force search in the array $Nodes$, indexed by all $O(|V(G)|^3)$ subsets of $V(G)$ of size at most three.

Finally, we prove that the algorithm is a 1.5-approximation. By Lemma 2.4, we have $\mathbf{tw}(G) \leq \max\{\mathbf{tw}(G_i) \mid i = 1, \ldots, m\}$. By Theorem 2.8, each G_i is a minor of G and therefore $\mathbf{tw}(G_i) \leq \mathbf{tw}(G)$ (since the class of graphs of treewidth at most

k is minor-closed). Thus, $\mathbf{tw}(G) = \max\{\mathbf{tw}(G_i) \mid i = 1, \ldots, m\}$ and from (2.7) we conclude that $\mathbf{tw}(G) \leq k \leq \frac{3}{2}\,\mathbf{tw}(G) + 1$ and the theorem follows. $\square$

Using the same approach as Theorem 2.18, one can prove a potentially stronger theorem:

Theorem 2.19. *If we can compute the treewidth of any planar graph in polynomial time, then we can compute the treewidth of any single-crossing-minor-free graph in polynomial time.*

Proof. We use the polynomial-time algorithm for computing treewidth of planar graphs in Step 2 of the algorithm described in the proof of Theorem 2.18. $\square$

2.5 Polynomial-time Approximation Schemes

2.5.1 General Schemes for Approximation on Special Classes of Graphs

A polynomial-time approximation scheme for a problem is a family $\{\mathcal{A}_\varepsilon\}$ of algorithms, where $\mathcal{A}_\varepsilon$ is a $(1+\varepsilon)$ approximation for the problem that runs in time polynomial in the length of its input (for fixed ε). Inherent in the design of many polynomial-time approximation schemes for NP-complete graph problems is the restriction of inputs to graph classes that guarantee additional structural properties. Early work in the area demonstrated the possibility of using planarity to obtain approximation schemes [132], later generalized to graphs without a fixed minor [9]. These approaches are impractical; a performance ratio of two for the independent set problem is achievable only for planar graphs of at least $2^{2^{400}}$ vertices [59].

Practical approximation schemes for planar graphs were developed by Baker [23], who formed a decomposition of G into overlapping k-outerplanar subgraphs. For any planar embedding, vertices can be put into layers by iteratively removing vertices on the outer face of the graph: vertices removed at the ith iteration are assigned to layer i. Since a k-outerplanar graph is decomposed into at most k layers, a k-outerplanar subgraph can be formed by removing vertices with layer number congruent to i mod k. Baker's technique immediately implies $(1 + 1/k)$-factor approximation algorithms for many problems that can be solved exactly on k-outerplanar graphs (e.g. maximum independent set, minimum dominating set, and minimum vertex cover), as it suffices to solve the problem exactly on each of the k subgraphs (one for each value of i) and return the best of the k results. Consider an optimal answer in the full graph. Since the sets of removed vertices partition the graph, one of these sets removes at most $1/k$ of the vertices in the answer, and the exact solution on the corresponding subgraph will be a $(1 + 1/k)$-approximation to the optimal answer for the full graph.

Chen [56] later generalized Baker's approach to form approximation algorithms of ratio $1 + 1/\log n$ for problems on $K_{3,3}$-minor-free graphs and K_5-minor-free graphs; due to the types of layers formed, these results were exclusively for maximization problems. Eppstein [87] showed that Baker's technique can be extended by replacing bounded outerplanarity with bounded local treewidth. As with k-outerplanar graphs, a wide range of NP-complete problems can be solved in linear time on graphs of bounded local treewidth. The decomposition by deleting every kth face is replaced by deleting every kth level of a breadth-first tree of G, keeping in mind the fact that that the treewidth of the resulting graphs is a function of k.

2.5.2 Approximation Schemes for Single-Crossing-Minor-Free Graphs

In this section we use the bound on local treewidth established in Theorem 2.10 to obtain polynomial-time approximation schemes. Among NP-optimization problems, we mainly focus on those problems which are also *hereditary*, namely, problems which determine a property that if valid for an input graph is also valid for any induced subgraph of the input. For a property π, the maximum induced subgraph problem MISP(π) is finding a maximum induced subgraph with the property; in the weighted version (WMISP(π)), the input graph has weights on its vertices and the goal is to find a maximum weight induced subgraph with the property. For example, we might search for an induced subgraph of maximum size that is chordal, acyclic, without cycles of a specified length, without edges, of maximum degree $r \geq 1$, bipartite or a clique [167]. For exact definitions of various NP-hard problems in this chapter, the reader is referred to Garey and Johnson's seminal book [99].

Yannakakis has shown that for many natural hereditary properties π, MISP(π) is NP-complete even when restricted to planar graphs [167]. Using the results of Section 2.3, we obtain approximation algorithms for both maximization and minimization problems such as the maximum independent set problem, the minimum vertex cover problem and the minimum dominating set problem on single-crossing-minor-free graphs.

Theorem 2.20. *For G a non-negative vertex-weighted single-crossing-minor-free graph excluding H, $k \geq 1$ an integer, and $Time_\pi(w, n)$ the nondecreasing worst-case running time of WMISP(π) over an n-vertex partial w-tree whose tree decomposition is given, the maximization problem WMISP(π) for a hereditary property π over G admits a polynomial-time approximation scheme of ratio $1 + 1/k$ with worst-case running time in $O(k \cdot |V|^4 + k \cdot Time_\pi(3(k-1) + c_H, |V|))$. The running time improves to $O(k \cdot |V| + k \cdot Time_\pi(3(k-1) + 4, |V|))$ for G $K_{3,3}$-minor-free and $O(k \cdot |V|^2 + k \cdot Time_\pi(3(k-1) + 4, |V|))$ for G K_5-minor-free.*

Proof. Our algorithm proceeds by creating k subgraphs of G, solving the problem on each of the subgraphs, and returning the best solution for any of the subgraphs as the solution for all of G. We make use of the locally bounded treewidth of G in order to specify layers from which the subgraphs are derived and to prove that each subgraph has bounded treewidth.

Given an assignment of vertices to layers numbered $1, 2, \ldots$ created by breadth-first search (layer i is all vertices at depth i), we use $L_{i,j}$ to denote the consecutive layers numbered $(j-1)k + i$ through $jk + i - 2$ for $1 \leq i \leq k$ and $j \geq 0$ where for convenience a layer is defined to be empty when its number is not between zero and the total number of layers. Furthermore, we let $\mathcal{L}_i = \bigcup_{j \geq 0} L_{i,j}$ and $G_i = G[\mathcal{L}_i]$. As neither $L_{i,j}$ nor $L_{i,j+1}$ contains the layer numbered $jk + i - 1$ and all edges appear between consecutive layers, there are no edges between $L_{i,j}$ and $L_{i,j+1}$. Moreover, as no vertices in layer $i-1$ appear in $\mathcal{L}_h$ for $h = i \bmod k$, each vertex appears in exactly $k-1$ of the $\mathcal{L}_i$'s or G_i's, a fact we will use later in the proof.

We next use the bound on the treewidth of each G_i to obtain Opt_i, the maximum weighted solution of WMISP(π) on each G_i, $1 \leq i \leq k$. In particular, we construct a tree decomposition of width $3(k-1)+c_H$ for each G_i by adding edges between tree decompositions for each $G[L_{i,j}]$, which in turn can be formed in $O(n^4)$ time using Theorem 2.14 ($O(n)$ for $K_{3,3}$-minor-free graphs or $O(n^2)$ time for K_5-minor-free graphs). The fact that the graphs $G[L_{i,j}]$ are disjoint means that the process of adding edges to form a single tree is straight-forward. As in Lemma 2.4, by the definition of $Time_\pi(w,n)$ as a nondecreasing function, since $|V(G_i)| \leq |V(G)|$, Opt_i can be determined in $Time_\pi(3(k-1)+c_H, |V(G)|)$. The running time is $Time_\pi(3(k-1)+4, |V(G)|)$ for G $K_{3,3}$-minor-free or K_5-minor-free.

Finally, we take Opt_m, the solution with maximum weight among the Opt_i's, as our solution for graph G, and prove the ratio bound by showing that $\frac{weight(Opt)}{weight(Opt_m)} \leq \frac{k}{k-1}$, or $k \cdot weight(Opt_m) \geq (k-1) \cdot weight(Opt)$, where Opt is the maximum weighted solution on graph G. By observing that $Opt_m \geq Opt_i$ for each value of i, we show that $k \cdot weight(Opt_m) \geq \sum_{i=1}^{k} weight(Opt_i)$. Since π is hereditary, $weight(Opt_i) \geq weight(Opt \cap \mathcal{L}_i)$, and hence $\sum_{i=1}^{k} weight(Opt_i) \geq \sum_{i=1}^{k} weight(Opt \cap \mathcal{L}_i)$. Finally, we recall that each vertex appears in exactly $k-1$ of the $\mathcal{L}_i$'s, from which we can conclude $\sum_{i=1}^{k} weight(Opt \cap \mathcal{L}_i) = (k-1) \cdot weight(Opt)$, as needed to conclude the proof that $k \cdot weight(Opt_m) \geq (k-1) \cdot weight(Opt)$.

The claimed running time follows immediately from the time to construct the tree decomposition, the time to solve WMISP(π) for each G_i, and the number of G_i's. □

Corollary 2.21. *For G a non-negative vertex-weighted single-crossing-minor-free graph excluding H, the maximum independent set problem admits a polynomial-time approximation scheme of ratio $1 + 1/k$ with running time $O(k \cdot 2^{3k} \cdot n + k \cdot n^4)$. The running time improves to $O(k \cdot 2^{3k} \cdot n)$ for G $K_{3,3}$-minor-free and $O(k \cdot 2^{3k} \cdot n + k \cdot n^2)$ for G K_5-minor-free.*

Proof. Using dynamic programming on a tree decomposition, this problem can be solved in $O(2^w \cdot n)$ time, over each n-vertex partial w-tree whose tree decomposition is given [8]. Thus the result follows from Theorem 2.20 for $Time_\pi(w, n) = O(2^w \cdot n)$. □

It is worth mentioning that our result can be applied to NP-minimization problems, e.g., the minimum vertex cover problem and the minimum dominating set problem. The main difference between these two results and the previous one is that the previous construction avoided overlap between the various sets $L_{i,j}$ of consecutive layers, in the minimization setting we need to enforce overlap in order to achieve the approximation guarantee. The ideas of the proofs of Theorems 2.22 and 2.23 follow ideas of Grohe [103] for general graphs of locally bounded treewidth, which are in fact Baker's ideas for planar graphs. Since there are not too many new ideas in the proofs of Theorems 2.22 and 2.23 below with respect to the proof of Theorem 2.20, we omit the proofs and refer the reader to [72] for their detailed proofs.

Theorem 2.22 ([72]). *For G a single-crossing-minor-free graph and any integer $k \geq 1$, the minimum weighted vertex cover problem admits a polynomial-time approximation scheme of ratio $1+1/k$ with worst-case running time $O(k \cdot 2^{3k} \cdot n + k \cdot n^4)$. The running time improves to $O(k \cdot 2^{3k} \cdot n)$ for G $K_{3,3}$-minor-free and $O(k \cdot 2^{3k} \cdot n + k \cdot n^2)$ for G K_5-minor-free.*

Theorem 2.23 ([72]). *For G a single-crossing-minor-free graph and any integer $k \geq 1$, the minimum weighted dominating set problem admits a polynomial-time approximation scheme of ratio $1+2/k$ with worst-case running time $O(k \cdot 4^{3k} \cdot n + k \cdot n^4)$. The running time improves to $O(k \cdot 4^{3k} \cdot n)$ for G $K_{3,3}$-minor-free and $O(k \cdot 4^{3k} \cdot n + k \cdot n^2)$ for G K_5-minor-free.*

Theorem 2.24. *For single-crossing-minor-free graphs, there are polynomial-time approximation algorithms whose solutions converge toward optimal as n increases for maximum independent set, minimum vertex cover and minimum dominating set.*

Proof. The running time of algorithms introduced in Corollary 2.21 and Theorems 2.22 and 2.23 is $O(c^k n + n^4)$ where k is a parameter and c is a constant. Now, by taking

$k = \lceil \log n \rceil$, we obtain polynomial-time approximation algorithms of ratio $1+1/(\log n)$ (or $1 + 2/(\log n)$ for dominating set). As both $1/(\log n)$ and $2/(\log n)$ decrease as n increases, the solutions converge toward optimal as n increases. □

Finally, it is worth mentioning that our results for single-crossing-minor-free graphs in this section have been generalized to all apex-minor-free graphs by a result of Demaine and Hajiaghayi [66] who showed that apex-minor-free graphs have linear local treewidth, i.e., the concepts of having bounded local treewidth and having linear local treewidth are equivalent in minor-closed graph families.

2.6 Concluding Remarks

In this chapter, we introduced the class of single-crossing-minor-free graphs, which contains $K_{3,3}$-minor-free graphs and K_5-minor-free graphs, generalizations of planar graphs, and demonstrated structural properties which gave rise to new algorithms. We showed that single-crossing-minor-free graphs have linear local treewidth and demonstrated how to obtain a tree decomposition of a fixed number of layers. Algorithms obtained using these properties include both approximation algorithms (e.g. a 1.5-approximation algorithm for treewidth and many polynomial-time approximation schemes) and fixed-parameter algorithms (that we demonstrate in Chapter 3).

Extensions to the structural results could include finding clique-sum characterizations of graphs such as graphs excluding a double-crossing graph (or a graph with a bounded number of crossings) as a minor. For each such class, polynomial-time constructions of the decompositions could be used for further algorithmic development.

Notice that the algorithm of Theorem 2.8 in Section 2.2.2 can serve as a general heuristic for the computation or approximation of treewidth in a graph when the resulting 3-connected components without strong 3-cuts are graphs whose treewidth can be computed or approximated efficiently. Additional approximation algorithms might include those which determine graph properties other than treewidth. As a consequence of Theorem 2.16, treewidth is a 1.5-approximation on branchwidth, which immediately implies the existence of a 2.25-approximation for branchwidth of single-crossing-minor-free graphs. It might be possible to use a clique-sum decomposition to obtain a better approximation or an exact algorithm for branchwidth. We believe that by using an approach similar to that described by Kezdy and McGuinness [118], it is possible to obtain a polylogarithmic parallel algorithm that constructs a series of clique-sum operations as in Theorem 2.8; details remain to be worked out.

We suspect that Baker's approach can be applied to obtain practical polynomial-time approximation schemes for other problems, such as variations on dominating sets [2] that have been solved on k-outerplanar graphs or graphs of bounded treewidth [46, 76, 2]; several results in this direction have been found recently [70]. By considering other NP-complete problems that have good algorithms for planar graphs and graphs of bounded treewidth, we may be able to extend the range of problems to which using clique-sum decomposition techniques may be applied; some results in this direction have already been found [74].

Chapter 3

Exponential Speedup of Fixed-Parameter Algorithms for Single-Crossing-Minor-Free Graphs

According to a 1998 survey book [112], there are more than 200 published research papers on solving domination-like problems on graphs. Because this problem is very hard and NP-complete even for special kinds of graphs such as planar graphs, much attention has focused on solving this problem on a more restricted class of graphs. It is well-known that this problem can be solved on trees [60] or even the generalization of trees, graphs of bounded treewidth [159]. The approximability of the dominating set problem has received considerable attention, but it is not known and it is not believed that this problem has constant-factor approximation algorithms on general graphs [22].

Downey and Fellows [83] introduced the concept of fixed-parameter tractability to handle NP-hardness. Unfortunately, according to this theory, it is very unlikely that the k-dominating set problem has an efficient fixed-parameter algorithm for general graphs. In contrast, this problem is fixed-parameter tractable on planar graphs. Alber et al. [2] demonstrated a solution to the planar k-dominating set in time $O(4^{6\sqrt{34k}}n)$. Indeed, this result was the first nontrivial result for the parameterized version of an NP-hard problem where the exponent of the exponential term grows sublinearly in the parameter. Recently, the running time of this algorithm was further improved to $O(2^{27\sqrt{k}}n)$ [116] and $O(2^{15.13\sqrt{k}}k + n^3 + k^4)$ [93]. One of the aims of this chapter is to generalize this result to nonplanar classes of graphs.

In this Chapter, similar to the approach of Alber et al., we prove that for a single-crossing graph H, the treewidth of any H-minor-free graph G having a k-dominating set is bounded by $O(\sqrt{k})$. As a result, we generalize current exponential speedup in fixed-parameter algorithms on planar graphs to other kinds of graphs by showing how we can solve the k-dominating set problem on any class of graphs excluding a single-crossing graph as a minor in time $O(4^{9.55\sqrt{k}}n^{O(1)})$. The genesis of our results lies in a result of Chapter 2 on obtaining the local treewidth of the aforementioned class of graphs.

Using the solution for the k-dominating set problem on planar graphs, Kloks et al. [50, 123, 107] and Alber et al. [2, 6] obtained exponential speedup in solving other problems such as vertex cover, independent set, clique-transversal set, kernels in digraph and feedback vertex set on planar graphs. In this chapter we also show how our results can be extended to these problems and many other problems such as variants of dominating set, edge dominating set, and a collection of vertex-removal problems.

This chapter is organized as follows. First, we introduce the preliminary definitions used throughout this chapter in Section 3.1. In Section 3.2, we prove two general theorems concerning the construction of tree decompositions of width $O(\sqrt{k})$ for these graphs, and finally we consider the design of fast fixed-parameter algorithms for them. In Section 3.2, we apply our general results to the k-dominating set problem, and in Section 3.3, we describe how this result can be applied to derive fast fixed-parameter algorithms for many different parameters. In Section 3.4, we prove some graph-theoretic results that provide a framework for designing fixed-parameter algorithms for a collection of vertex-removal problems. In Section 3.5, we give some further extensions of our results to graphs with linear local treewidth. We end with some conclusions and open problems in Section 3.6.

3.1 Background

3.1.1 Preliminaries

For generalizations of algorithms on undirected graphs to directed graphs, we consider underlying graphs of directed graphs. The *underlying graph* of a directed graph H is the undirected graph G in which $V(G) = V(H)$ and $\{u, v\} \in E(G)$ if and only if $(u, v) \in E(H)$ or $(v, u) \in E(H)$.

Let s be an integer where $0 \leq s \leq 3$ and $\mathcal{C}$ be a finite set of graphs. We say that a graph class $\mathcal{G}$ is a *clique-sum class* if any of its graphs can be constructed by a sequence of i-sums ($i \leq s$) applied to planar graphs and graphs in $\mathcal{C}$. We call a graph *clique-sum* if it is a member of a clique-sum class. We call the pair $(\mathcal{C}, s)$ the *defining pair* of $\mathcal{G}$ and we call the maximum treewidth of graphs in $\mathcal{C}$ the *base* of $\mathcal{G}$ and the *base* of graphs in $\mathcal{G}$. A series of k-sums (not necessarily unique) which generate a clique-sum graph G are called *a decomposition of G into clique-sum operations.*

According to the (nonalgorithmic) result of [142], if $\mathcal{G}$ is the class of graphs excluding a single crossing graph H then $\mathcal{G}$ is a clique-sum class with defining pair $(\mathcal{C}, s)$ where the base of $\mathcal{G}$ is bounded by a constant c_H depending only on H. In particular, if $H = K_{3,3}$, the defining pair is $(\{K_5\}, 2)$ and $c_H = 4$ [164] and if $H = K_5$ then the defining pair is $(\{V_8\}, 3)$ and $c_H = 4$ [164].

We call a clique-sum graph class $\mathcal{G}$ *α-recognizable* if there exists an algorithm that for any graph $G \in \mathcal{G}$ outputs in $O(n^\alpha)$ time a sequence of clique-sums of graphs of total size $O(|V(G)|)$ that constructs G. We call a graph *α-recognizable* if it belongs in some α-recognizable clique-sum graph class.

One of the ingredients of our results is the following constructive version of the result in [142], which has been proved in Chapter 2.

Theorem 3.1 ([72]). *For any graph G excluding a single-crossing graph H as a minor, we can construct in $O(n^4)$ time a series of clique-sum operations $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is a minor of G and is either a planar graph or a graph of treewidth at most c_H. Here each $\oplus$ is a 0-, 1-, 2- or 3-sum.*

In the remainder of the chapter we assume that c_H is the smallest integer for which Theorem 3.1 holds. Notice that, according to the terminology introduced before, any graph class excluding a single-crossing graph as a minor is a 4-recognizable clique-sum graph class. As particular cases of Theorem 3.1 we mention that $K_{3,3}$-minor-free graphs are 1-recognizable [20] and K_5-minor-free graphs are 2-recognizable [118]. For more examples of graph classes that can be characterized by clique-sum decompositions, see the work of Diestel [77, 78].

A *parameterized graph class* (or just *graph parameter*) is a family $\mathcal{F}$ of classes $\{\mathcal{F}_i, i \geq 0\}$ where $\bigcup_{i\geq 0} \mathcal{F}_i$ is the set of all graphs and for any $i \geq 0$, $\mathcal{F}_i \subseteq \mathcal{F}_{i+1}$. Given two parameterized graph classes $\mathcal{F}^1$ and $\mathcal{F}^2$ and a natural number $\gamma \geq 1$ we say that $\mathcal{F}^1 \preccurlyeq_\gamma \mathcal{F}^2$ if for any $i \geq 0$, $\mathcal{F}^1_i \subseteq \mathcal{F}^2_{\gamma \cdot i}$.

In the rest of this chapter, we identify a parameterized problem with the *parameterized graph class* corresponding to its "yes" instances.

Theorem 3.2. *Let $\mathcal{G}$ be an α_1-recognizable clique-sum graph class with base c and let $\mathcal{F}$ be a parameterized graph class. In addition, we assume that each graph in $\mathcal{G}$ can be constructed using i-sums where $i \leq s \leq 3$. Suppose also that there exist two positive real numbers β_1, β_2 such that:*

(1) For any $k \geq 0$, planar graphs in $\mathcal{F}_k$ have treewidth at most $\beta_1\sqrt{k} + \beta_2$ and such a tree decomposition can be found in $O(n^{\alpha_2})$ time.

(2) For any $k \geq 0$ and any $i \leq s$, if $G_1 \oplus_i G_2 \in \mathcal{F}_k$ then $G_1, G_2 \in \mathcal{F}_k$

Then, for any $k \geq 0$, the graphs in $\mathcal{G} \cap \mathcal{F}_k$ all have treewidth $\leq \max\{\beta_1\sqrt{k} + \beta_2, c\}$ and such a tree decomposition can be constructed in $O(n^{\max\{\alpha_1,\alpha_2\}} + (\sqrt{k})^s \cdot n)$ time.

Proof. Let $G \in \mathcal{G} \cap \mathcal{F}_k$ and assume that $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is either a planar graph or a graph of treewidth at most c. We use induction on m, the number of G_i's. For $m = 1$, $G = G_1$ is either a planar graph that from (1) has treewidth at most $\beta_1\sqrt{k} + \beta_2$ or a graph of treewidth at most c. Thus the basis of the induction is true for both cases. We assume the induction hypothesis is true for $m = h$, and we prove the hypothesis for $m = h + 1$. Let

$G' = G_1 \oplus G_2 \oplus \cdots \oplus G_h$ and $G'' = G_{h+1}$. Thus $G = G' \oplus G''$. By (2), both G' and G'' belong in $\mathcal{F}_k$. By the induction hypothesis, $\mathbf{tw}(G') \leq \max\{\beta_1\sqrt{k} + \beta_2, c\}$ and from (1) $\mathbf{tw}(G'') \leq \max\{\beta_1\sqrt{k} + \beta_2, c\}$. The proof, for $m = h + 1$, follows from this fact and Lemma 2.4.

To construct a tree decomposition of the aforementioned width, first we construct a tree decomposition of width at most $\beta_1\sqrt{k} + \beta_2$ for each planar graph in $O(n^{\alpha_2})$ time. We also note that using Bodlaender's algorithm [32], we can obtain a tree decomposition of width c for any graph of treewidth at most c in linear time (the hidden constant only depends on c). Then having tree decompositions of G_i's, $1 \leq i \leq m$, in the rest of the algorithm, we glue together the tree decompositions of G_i's using the construction given in the proof of Lemma 2.4. To this end, we introduce an array *Nodes* indexed by all subsets of $V(G)$ of size at most s. In this array, for each subset whose elements form a clique, we specify a node of the tree decomposition which contains this subset. We note that for each clique C in G_i, there exists a node z of $TD(G)$ such that all vertices of C appear in the bag of z [37]. This array is initialized as part of a preprocessing stage of the algorithm. Now, for the $\oplus$ operation between $G_1 \oplus \cdots \oplus G_h$ and G_{h+1} over the join set W, using array *Nodes*, we find a node α in the tree decomposition of $G_1 \oplus \cdots \oplus G_h$ whose bag contains W. Because we have the tree decomposition of G_{h+1}, we can find the node α' of the tree decomposition whose bag contains W by brute force over all subsets of size at most s of bags. Simultaneously, we update array *Nodes* by subsets of $V(G)$ which form a clique and appear in bags of the tree decomposition of G_{h+1}. Then we add an edge between α and α'. As the number of nodes in a tree decomposition of G_{h+1} is in $O(|V(G_{h+1})|)$ and each bag has size at most $O(\sqrt{k})$ (and thus there are at most $O((\sqrt{k})^s)$ choices for a subset of size at most s), this operation takes $O((\sqrt{k})^s|V(G_{h+1})|)$ time for G_{h+1}.

The claimed running time follows from the time required to determine a set of clique-sum operations, the time required to construct tree decompositions, the time needed for gluing tree decompositions together and the fact that $\sum_{i=1}^{m} |V(G_i)| = O(|V(G)|)$. $\square$

Notice that condition (2) of Theorem 3.2 is not necessary when $\mathcal{G}$ excludes a single-crossing graph and $\mathcal{F}$ is closed under taking of minors. Indeed, from Theorem 3.1, we have that in the sequence of operations $G = G_1 \oplus G_2 \oplus \ldots \oplus G_m$, each G_i is a minor of G and therefore, if $G \in F_k$ then each G_i is also a member of F_k. We resume this observation to the following.

Theorem 3.3. *Let $\mathcal{G}$ be the class of graphs excluding some single-crossing graph H as a minor and let $\mathcal{F}$ be any minor-closed parameterized graph class. Suppose that there exist real numbers $\beta_0 \geq 4, \beta_1$ such that any planar graph in $\mathcal{F}_k$ has treewidth at most $\max\{\beta_1\sqrt{k} + \beta_0, c_H\}$ and such a tree decomposition can be found in $O(n^\alpha)$. Then graphs in $\mathcal{G} \cap \mathcal{F}_k$ all have treewidth $\leq \beta_1\sqrt{k} + \beta_0$ and such a tree decomposition can be constructed in $O(n^{\max\{\alpha,4\}})$ time.*

Theorem 3.4. *Let $\mathcal{G}$ be a graph class and let $\mathcal{F}$ be some parameterized graph class. Suppose also for some positive real numbers $c, \alpha_1, \alpha_2, \beta_1, \beta_2, \delta$ the following hold:*

(1) For any $k \geq 0$, the graphs in $\mathcal{G} \cap \mathcal{F}_k$ all have treewidth $\leq \max\{c, \beta_1\sqrt{k} + \beta_2\}$ and such a tree decomposition can be decided and constructed (if it exists) in $O(n^{\alpha_2})$ time. We also assume testing membership in $\mathcal{G}$ takes $O(n^{\alpha_1})$ time.

(2) Given a tree decomposition of width at most w of a graph, there exists an algorithm deciding whether the graph belongs in $\mathcal{F}_k$ in $O(\delta^w n)$ time.

Then there exists an algorithm deciding in $O(\delta^{\max\{c,\beta_1\sqrt{k}+\beta_2\}} + n^{\max\{\alpha_1,\alpha_2\}})$ time whether an input graph G belongs in $\mathcal{G} \cap \mathcal{F}_k$.

Proof. First, we can test membership in $\mathcal{G}$ in $O(n^{\alpha_1})$ time. Then we can apply the algorithm from (1) and (assuming success) supply the resulting tree decomposition to the algorithm from (2). □

3.2 Fixed-Parameter Algorithms for Dominating Set

In this section, we describe some of the consequences of Theorems 3.2 and 3.4 on the design of efficient fixed-parameter algorithms for a collection of parameterized problems where their inputs are clique-sum graphs.

A *dominating set* of a graph G is a set of vertices of G such that each of the rest of vertices has at least one neighbor in the set. We represent the *k-dominating set* problem with the parameterized graph class $\mathcal{DS}$ where $\mathcal{DS}_k$ contains graphs which have a dominating set of size $\leq k$. Our target is to show how we can solve the k-dominating set problem on clique-sum graphs, where H is a single-crossing graph, in time $O(c^{\sqrt{k}}n^{O(1)})$ instead of the current algorithms which run in time $O(c^k n^{O(1)})$ for some constant c. By this result, we extend the current exponential speedup in designing algorithms for planar graphs [6] to a more generalized class of graphs. In fact, planar graphs are both $K_{3,3}$-minor-free and K_5-minor-free graphs, where both $K_{3,3}$ and K_5 are single-crossing graphs.

According to the result of [116] condition (1) of Theorem 3.2 is satisfied for $\beta_1 = 15.6$, $\beta_2 = 50$, and $\alpha_2 = 1$. Moreover, from [93], condition (1) is also satisfied for $\beta_1 = 9.55$, $\beta_2 = 0$ and $\alpha_2 = 4$.

The next lemma shows that condition (2) of Theorem 3.2 also holds.

Lemma 3.5. *If $G = G_1 \oplus_m G_2$ has a k-dominating set, then both G_1 and G_2 have dominating sets of size at most k.*

Proof. Let the k-dominating set of G be S and W be the join set of $G_1 \oplus_k G_2$. W.l.o.g. we show that G_1 has a dominating set of size k. If $S_1 = S \cap V(G_1)$ is a dominating set for G_1 then the result immediately follows, otherwise there exists vertex $w \in V(G_1)$ which is dominated by a vertex $v \in V(G_2) - V(G_1)$. One can observe that all such vertices w are in W. Because $v \in S$, but $v \notin S_1$, set $S_1' = S_1 + \{w\}$ has at most k vertices and because W is a clique in G_1, S_1' is a dominating set of size at most k in G_1. $\square$

Let $\mathcal{G}$ be any α-recognizable clique-sum class. Now by applying Theorem 3.2 for $\beta_1 = 9.55$, $\beta_2 = 0$, $\alpha_1 = \alpha$, and $\alpha_2 = 4$ we have the following.

Theorem 3.6. *If $\mathcal{G}$ is an α-recognizable clique-sum class of base c, then any member G of $\mathcal{G}$ with a dominating set of size at most $\leq k$ has treewidth at most $\max\{c, 9.55\sqrt{k}\}$ and the corresponding tree decomposition of G can be constructed in $O(n^{\max\{\alpha,4\}})$ time.*

From Theorem 3.6, we get that condition (1) of Theorem 3.4 is satisfied for $\beta_1 = 9.55$, $\beta_2 = 0$, $\alpha_2 = \max\{\alpha, 4\}$, and $\alpha_2 = 4$. The main result in [8] shows that for the graph parameter $\mathcal{DS}$ condition (2) of Theorem 3.4 is also satisfied for $\delta = 4$. We conclude with the following.

Theorem 3.7. *There is an algorithm that in $O(4^{9.55\sqrt{k}}n + n^{\max\{\alpha,4\}})$ time solves the k-dominating set problem for any α-recognizable clique-sum graph of base c.*[1]

Corollary 3.8. *There is an algorithm that solves the k-dominating set problem for any graph class excluding some single crossing graph as a minor in $O(4^{9.55\sqrt{k}}n + n^4)$ time.*

For the special cases of K_5-minor-free graphs and $K_{3,3}$-minor-free graphs, we may apply Theorem 3.2 for $\beta_1 = 15.6$, $\beta_2 = 50$, and $\alpha_2 = 1$ and derive the following.

Corollary 3.9. *There is an algorithm that solves the k-dominating set problem for any K_5-minor-free graph in $O(4^{15.6\sqrt{k}+50}n + n^2)$ time and for any $K_{3,3}$-minor-free graph in $O(4^{15.6\sqrt{k}+50}n)$ time.*

3.3 Algorithms for Parameters Bounded by the Dominating-Set Number

We provide a general methodology for deriving fast fixed-parameter algorithms in this section. First, we consider the following theorem which is an immediate consequence of Theorem 3.4.

Theorem 3.10. *Let $\mathcal{G}$ be a graph class and let $\mathcal{F}^1$, $\mathcal{F}^2$ be two parameterized graph classes where $\mathcal{F}^1 \preccurlyeq_\gamma \mathcal{F}^2$ for some natural number $\gamma \geq 1$. Suppose also that there exist positive real numbers $\alpha_1, \alpha_2, \beta_1, \beta_2, \delta$ such that:*

(1) For any $k \geq 0$, the graphs in $\mathcal{G} \cap \mathcal{F}^2_k$ all have treewidth $\leq \beta_1\sqrt{k} + \beta_2$ and such a tree decomposition can be decided and constructed (if it exists) in $O(n^{\alpha_2})$ time. We also assume testing membership in $\mathcal{G}$ takes $O(n^{\alpha_1})$ time.

[1] In the rest of this chapter, we assume that constants, e.g. c, are small and they do not appear in the powers, because they are absorbed into the O notation.

(2) There exists an algorithm deciding whether a graph of treewidth $\leq w$ belongs in $\mathcal{F}_k^1$ in $O(\delta^w n)$ time.

Then

(1) For any $k \geq 0$, the graphs in $\mathcal{G} \cap \mathcal{F}_k^1$ all have treewidth at most $\beta_1\sqrt{\gamma k} + \beta_2$ and such a tree decomposition can be constructed in $O(n^{\alpha_2})$ time.

(2) There exists an algorithm deciding in $O(\delta^{\beta_1\sqrt{\gamma k}+\beta_2} + n^{\max\{\alpha_1,\alpha_2\}})$ time whether an input graph G belongs in $\mathcal{G} \cap \mathcal{F}_k^1$.

Proof. Consequence (1) follows immediately from the definition of $\preccurlyeq_\gamma$. Consequence (2) follows from Theorem 3.4. □

The idea of our general technique is given by the following theorem that is a direct consequence of Theorems 3.6 and 3.10.

Theorem 3.11. *Let $\mathcal{F}$ be a parameterized graph class satisfying the following two properties:*

(1) It is possible to check membership in $\mathcal{F}_k$ of a graph G of treewidth at most w in $O(\delta^w n)$ time for some positive real number δ.

(2) $\mathcal{F} \preccurlyeq_\gamma \mathcal{DS}$.

Then

(1) Any clique-sum graph G of base c in $\mathcal{F}_k$ has treewidth at most $\max\{9.55\sqrt{\gamma k} + 8, c\}$.

(2) We can check whether an input graph G is in $\mathcal{F}_k$ in $O(\delta^{9.55\sqrt{\gamma k}}n + n^{\max\{\alpha,4\}})$ on an α-recognizable clique-sum graph of base c.

In what follows we explain how Theorem 3.11 applies for a series of graph parameters. In particular, we explain why Conditions (1) and (2) are satisfied for each problem.

3.3.1 Variants of the Dominating Set Problem

A *k-dominating set with property* Π on an undirected graph G is a k-dominating set D of G which has the additional property Π and the *k-dominating set with property* Π *problem* is the task to decide, given a graph $G = (V, E)$, a property Π, and a positive integer k, whether or not there is a k-dominating set with property Π. Some examples of this type of problems, which are mentioned in [2, 159, 160], are the *k-independent dominating set* problem, the *k-total dominating set* problem, the *k-perfect dominating set* problem, the *k-perfect independent dominating set* problem also known as *k-perfect code* and the *k-total perfect dominating set* problem. For each Π, we denote the corresponding dominating set problem by $\mathcal{DS}^{\Pi}$.

Another variant is *the weighted dominating set problem* in which we have a graph $G = (V, E)$ together with an integer weight function $w : V \to \mathbb{N}$ with $w(v) > 0$ for all $v \in V$. The weight of a vertex set $D \subseteq V$ is defined as $w(D) = \sum_{v \in D} w(v)$. A k-weighted dominating set D of an undirected graph G is a dominating set D of G with $w(D) \leq k$. The *k-weighted dominating set* problem is the task of deciding whether or not there exits a k-weighted dominating set. We use the parameterized class $\mathcal{WDS}$ to denote the k-weighted dominating set problem.

Condition (1) of Theorem 3.11 holds for $\delta = 4$ because of the following.

Theorem 3.12 ([2]). *If a tree decomposition of width w of a graph is known, then a solution to $\mathcal{DS}^{\Pi}$ or to $\mathcal{WDS}$ can be determined in at most $O(4^w \cdot n)$ time.*

Clearly, $\mathcal{DS}^{\Pi} \preccurlyeq_1 \mathcal{DS}$ and $\mathcal{WDS} \preccurlyeq_1 \mathcal{DS}$ and Condition (2) also holds. Therefore Theorem 3.11 holds for $\gamma = 1$ and $\delta = 4$ for $\mathcal{DS}^{\Pi}$ and $\mathcal{WDS}$.

Another related problem is the Y-domination problem ($\mathcal{DS}^{Y}$) introduced in [25].

Definition 3.13. *Let Y be a finite set of integers. A* Y-domination *is an assignment $f : V \to Y$ such that for each vertex x, $f(N[x]) = \sum_{v \in N[x]} f(x) \geq 1$ where $N[x]$ stands for the neighborhood of x including x itself. An* efficient Y-domination *is an assignment f with $f(N[x]) = 1$ for all vertices $x \in V$. The* value *of a Y-domination f is $|\{x \mid f(x) > 0\}|$. The weight of a Y-domination is $\sum_{x \in V} f(x)$. Two Y-dominations are* equivalent *if they have the same closed neighborhood sum at every vertex. The*

Y-domination problem *asks whether the input graph G has an efficient Y-domination of value at most k.*

Using the generalized dynamic programming approach, Cai and Kloks [122] presents an algorithm which runs in time $O(|Y|^w n)$ to decide whether a graph G of treewidth at most w has an efficient Y-domination of value at most k. It is worth mentioning that, according to Bange et al. [25], a graph G has an efficient Y-domination if and only if all equivalent Y-dominations have the same weight, and thus there is no need to worry about the actual weight of an efficient Y-domination. Therefore, we have that Condition (1) of Theorem 3.11 holds for $\delta = |Y|$.

One can easily see that for Y-domination f of a graph $G = (V, E)$, $D = \{x | f(x) > 0\}$ is a dominating set, because each vertex x has at least one vertex with a positive number assigned to it in $N[x]$. Thus if f is a Y-domination of G with value at most k, then G also has a dominating set of size k. Therefore, $\mathcal{DS}^Y \preccurlyeq_1 \mathcal{DS}$ and Condition (2) holds as well. Theorem 3.11 applies for $\gamma = 1$ and $\delta = |Y|$.

3.3.2 Vertex Cover

The k-vertex cover problem ($\mathcal{VC}$) asks whether there exists a subset C of at most k vertices such that every edge of G has at least one endpoint in C. This problem is one of the most popular problems in combinatorial optimization.

A great number of researchers believe that there is no polynomial time approximation algorithm achieving an approximation factor strictly smaller than $2 - \varepsilon$, for a positive constant ε, unless $P = NP$. Currently, the best known lower bound for this factor is 1.36 [82] and the best upper bound is 2 which can be obtained easily. The best current fixed-parameter tractable algorithm has time $O(1.271^k + k|V|)$ [55]. In this section, we present an exponentially faster algorithm for this problem on clique-sum graphs.

Without loss of generality, we can restrict our attention to graphs with no vertex of degree zero. One can observe that if a graph G has a vertex cover of size k, then it has also a k-dominating set. Therefore $\mathcal{VC} \preccurlyeq_1 \mathcal{DS}$ and condition (1) of Theorem 3.11

holds. Moreover, Condition (2) holds because we can solve the vertex cover problem in time $O(2^w)$ if we know the tree decomposition of width w of a graph G [6]. Therefore, Theorem 3.11 applies for $\gamma = 1$ and $\delta = 2$ for the k-vertex problem.

A simple standard reduction to the problem kernel due to Buss and Goldsmith [47] is as follows: Each vertex of degree greater than k must be in the vertex cover of size k, because otherwise, not all edges can be covered. Thus we can obtain a subgraph G' of G which has at most k^2 edges and at most $k^2 + k$ vertices and k' is obtained from k reduced by the number of vertices of degree more than k. Chen et al. [55] showed that in Buss and Goldsmith's approach one can even obtain a problem kernel with at most $2k$ vertices in $O(nk + k^3)$ time. Thus, using this result with the consequence (2) of Theorem 3.11 for $\mathcal{VC}$, we obtain the following result.

Theorem 3.14. *There exists an algorithm which decides the k-vertex cover problem in $O(2^{9.55\sqrt{k}}k + kn + k^3 + n^{\max\{\alpha,4\}})$ time on an α-recognizable clique-sum graph.*

3.3.3 Edge Dominating Set

Another related problem is the edge dominating set problem $\mathcal{EDS}$ that given a graph G asks for a set $E' \subseteq E$ of k or fewer edges such that every edge in E shares at least one endpoint with some edge in E'. Again without loss of generality we can assume that graph G has no vertex of degree zero.

One can observe that if a graph G has a k-edge dominating set E', we can obtain a vertex cover of size $2k$ by including both end-points of each edge $e \in E'$. This means that $\mathcal{EDS} \preccurlyeq_2 \mathcal{VC}$. In the previous section we showed that $\mathcal{VC} \preccurlyeq_1 \mathcal{DS}$ therefore, the Condition (2) of Theorem 3.11 holds for $\mathcal{EDS}$ when $\gamma = 2$. Condition (1) holds because the edge dominating set problem can be solved in $c_{\mathsf{eds}}^{w} n$ [29, 23] (where c_{eds} is a small constant) on a tree decomposition of width w for a graph G. We conclude that Theorem 3.11 applies for $\gamma = 2$ and $\delta = c_{\mathsf{eds}}$.

Theorem 3.15. *We can find a k-edge dominating set in $O({c_{eds}}^{9.55\sqrt{2k}}n + n^{\max\{\alpha,4\}})$ time on an α-recognizable clique-sum graph.*

3.3.4 Clique-Transversal Set

A *clique-transversal set* of a connected graph G is a subset of vertices intersecting all the maximal cliques of G [24, 49, 13, 106]. Because the vertex cover problem is NP-complete even restricted to triangle-free planar graphs [50, 163], the clique-transversal problem remains NP-complete on clique-sum graphs. The *k-clique transversal* problem $\mathcal{CT}$ asks whether the input graph has a clique-transversal set of size $\leq k$.

If a graph G has a k-clique-transversal, then it has a dominating set of size at most k, because every vertex of G is contained in at least one maximal clique. This implies that $\mathcal{CT} \preccurlyeq_1 \mathcal{DS}$ and Condition (2) of Theorem 3.11 holds for $\gamma = 1$. Using the general dynamic programming technique, we can solve the k-clique-transversal problem on a graph G of treewidth at most w in $O(c_{\mathsf{ct}}^w n)$ for some constant c_{ct}. (The approach is very similar to Chang et al. [50].) Therefore, Theorem 3.11 applies for $\gamma = 1$ and $\delta = c_{\mathsf{ct}}$.

Theorem 3.16. *We can find a k-clique-transversal set in* $O({c_{ct}}^{9.55\sqrt{k}} n + n^{\max\{\alpha,4\}})$ *time on an* α*-recognizable clique-sum graph.*

3.3.5 Maximal Matching

A *matching* in a graph G is a set E' of edges without common endpoints. A matching in G is *maximal* if there is no other matching in G containing it. The k-maximal matching problem $\mathcal{MM}$ asks whether an input graph G has a maximal matching of size $\leq k$.

Let E' be the edges of a maximal matching of G. Notice that the set of endpoints of the edges in E' is a dominating set of G. Therefore $\mathcal{MM} \preccurlyeq_2 \mathcal{DS}$ and the Condition (2) of Theorem 3.11 holds. Condition (1) holds because the problem can be solved in $c_{\mathsf{mm}}^w n$ [29] on a tree decomposition of width w for a graph G. Hence Theorem 3.11 gives the following result.

Theorem 3.17.

(1) Any clique-sum graph of base c with a minimum maximal marching of size k has treewidth $\leq 9.55\sqrt{2k} + \max\{8, c\}$.

(2) One can decide whether an α-recognizable clique-sum graph G has a minimum maximal matching of size at most k in $O(c_{\mathsf{mm}}^{9.55\sqrt{2k}}n + n^{\max\{\alpha,4\}})$ time.

3.3.6 Kernels in Digraphs

A set S of vertices in a digraph $D = (V, A)$ is a kernel if S is independent and every vertex in $V - S$ has an out-neighbor in S. It has been shown that the problem of deciding whether a digraph has a kernel is NP-complete [99]. Franenkel [95] showed that the kernel problem remains NP-complete even for planar digraphs D with in-degree and outdegree at most 2 and total degree at most 3. The k-kernel problem $\mathcal{KER}$ asks whether a graph has a kernel of size k. Moreover, we define the co-$\mathcal{KER}$ problem as the one asking whether an n-vertex graph has a kernel of size $n - k$.

Here, we again observe that if a digraph D has a kernel of size at most k, then its underlying graph G has a dominating set of cardinality at most k. Also for a connected digraph $D = (V, A)$ and kernel K, $V - K$ is a dominating set in the underlying graph of D. Resuming these two facts we have $\mathcal{KER} \preccurlyeq_1 \mathcal{DS}$ and co-$\mathcal{KER} \preccurlyeq_1 \mathcal{DS}$ and Condition (2) of Theorem 3.11 holds for both problems. We note that a slight variation of Condition (1) also holds because Gutin et al. [107] gives an $O(3^w kn)$ time algorithm solving the k-kernel problem on graphs of treewidth at most w using the general dynamic programming approach. The straightforward adaptation of Theorem 3.11 to this variation of Condition (1) gives the following.

Theorem 3.18.

(1) Any clique-sum graph of base c that has a kernel of size k or $n - k$ has treewidth $\leq 9.55\sqrt{k} + \max\{8, c\}$.

(2) One can decide whether an α-recognizable clique-sum graph G of base c has a kernel of size k in $O(3^{9.55\sqrt{k}}nk + n^{\max\{\alpha,4\}})$ time.

(3) One can decide whether an α-recognizable clique-sum graph G of base c has a kernel of size $n - k$ in $O(3^{9.55\sqrt{k}}n(n-k) + n^{\max\{\alpha,4\}})$ time.

3.4 Fixed-Parameter Algorithms for Vertex-Removal Problems

In this section, we present general results allowing the construction of $O(c^{\sqrt{k}}n)$-time algorithms for a collection of vertex-removal problems. To this end, we start with some definitions. For any graph class $\mathcal{G}$ and any nonnegative integer k the graph class k-almost$(\mathcal{G})$ contains any graph $G = (V, E)$ where there exists a subset $S \subseteq V(G)$ of size at most k such that $G[V - S] \in \mathcal{G}$. We note that using this notation if $\mathcal{G}$ contains all the edgeless graphs or forests then k-almost$(\mathcal{G})$ is the class of graphs with vertex cover $\leq k$ or feedback vertex set $\leq k$.

We define $\mathcal{T}_r$ to be the class of graphs with treewidth $\leq r$. It is known that, for $1 \leq i \leq 2$, $\mathcal{T}_i$ is exactly the class of K_{i+2}-minor-free graphs (see e.g. [33]). We now present a series of consequences of Theorem 3.3 for solving a collection of vertex-removal problems on classes of graphs excluding a single-crossing graph as a minor. First, we need the following combinatorial lemma.

Lemma 3.19. *Planar graphs in k-almost$(\mathcal{T}_2)$ have treewidth $\leq 9.55\sqrt{k}$. Moreover, such a tree decomposition can be found in $O(n^4)$ time.*

Proof. Our target is to prove that planar graphs in k-almost$(\mathcal{T}_2)$ are subgraphs of planar graphs in $\mathcal{DS}_k$ and the result will follow from the fact that from [93], condition (1) of Theorem 3.2 is also satisfied for $\beta_1 = 9.55$, $\beta_2 = 0$ and $\alpha_2 = 4$.

Let G be a planar graph and S be a set of $\leq k$ vertices in G where $G[V - S]$ is K_4-minor-free. Using Lemma 2.4, we can assume that G is a biconnected graph. In addition, because k-almost$(\mathcal{T}_2)$ is a minor closed class, we can assume that G does not have a 2-cut (a cut of size 2). In fact, if G has a 2-cut $\{u, v\}$, each of the connected components of $G - \{u, v\}$ plus edge $\{u, v\}$ is a minor of G and thus by induction, we can assume that they satisfy the conditions of the theorem. Then using Lemma 2.4, we can glue the corresponding tree-decompositions together and obtain the desired result for G. All these operations take at most $O(n^3)$ time.

Therefore, in the rest of the proof we assume that G does not have 1- or 2-

cuts. A consequence of this is that all the vertices of G have degree at least 3. Another consequence is that two faces of G can have in common either a vertex or an edge (otherwise, a 2-cut appears). Consider any planar embedding of G. We call a face of this embedding *exterior* if it contains a vertex of S, otherwise we call it interior. For each exterior face choose a vertex in S and connect it with the rest of its vertices. We call the resulting graph H and we note that (a) G is a subgraph of H, (b) $H[V-S] = G[V-S]$, and (c) all the vertices of the exterior faces of H are dominated by some vertex in S. We claim that S is a dominating set of H. Suppose, towards a contradiction, that there is a vertex v that is not dominated by S. From (c) we can assume that all of the faces containing v are interior. Let H' be the graph induced by the vertices of these faces. As they are all interior, H' should be a subgraph of $H[V-S]$. Let $(x_1, \ldots, x_q, x_1)$ be a cyclic order of the neighbors of v and notice that $q \geq 3$. Let also F_i be the face of H containing the vertices $x_i, v, x_{\mathsf{next}(i)}$, $1 \leq i \leq q$, where $\mathsf{next}(i) = (i+1) \mod q+1$. We note that all these faces are pairwise distinct otherwise v will be a 1-cut for H and G. Let P_i be the path connecting x_i and $x_{\mathsf{next}(i)}$ in H' avoiding v and containing only vertices of F_i. Recall now that two faces of H have either v or an edge containing v in common. Therefore, it is impossible two paths P_i, $P_j, i \neq j$, share an internal vertex. This implies that the contraction of all the edges but one of each of these paths transforms H' to a wheel W_q that, as $q \geq 3$, can be further contracted to a K_4 (a weel W_q is the graph constructed taking a cycle of length q and connecting all its vertices with a new vertex v). As H' is a subgraph of the graph $H[V-S]$ (b) implies that $G[V-S]$ contains a K_4, and this is a contradiction. As now S is a dominating set for H the treewidth of H is at most $9.55\sqrt{k}$. From (a) we have that G is a subgraph of a planar graph in $\mathcal{DS}_k$ and this completes the proof of the theorem. □

As mentioned before, k-almost$(\mathcal{T}_2)$ is a minor closed graph class. Moreover, if $\mathcal{O} \subseteq \mathcal{T}_2$, then for any k, k-almost$(\mathcal{O}) \subseteq k$-almost$(\mathcal{T}_2)$. Using now Theorem 3.3 we conclude the following general result.

Theorem 3.20. *Let $\mathcal{O}$ be any class of graphs with treewidth ≤ 2 and let $\mathcal{G}$ the class of* *graphs excluding some single-crossing graph H as a minor. Then the following hold.*

(1) For any $k \geq 0$, all graphs in k-almost($\mathcal{O}$) that also belong to $\mathcal{G}$ have treewidth $\leq \max\{9.55\sqrt{k}, c_H\}$. Moreover, the corresponding tree decomposition can be found in $O(n^4)$ time.

(2) Suppose also that there exists an $O(\delta^w n)$ algorithm that decides whether a given graph belongs in k-almost($\mathcal{O}$) for graphs of treewidth at most w. Then, one can decide whether a graph in $\mathcal{G}$ belongs in k-almost($\mathcal{O}$) in $O(\delta^{9.55\sqrt{k}}n + n^4)$ time.

If $\{O_1, \ldots, O_r\}$ is a finite set of graphs, we denote by minor-excl$(O_1, \ldots, O_r)$ the class of graphs that are O_i-minor-free for all $i = 1, \ldots, r$.

As examples of problems for which Theorem 3.20 can be applied, we mention the problems of checking whether a graph, after removing k vertices, is *edgeless* ($\mathcal{G} = \mathcal{T}_0$), or has *maximum degree* ≤ 2 ($\mathcal{G} =$ minor-excl$(K_{1,3})$), or becomes a *a star forest* ($\mathcal{G} =$ minor-excl(K_3, P_3)), or a *caterpillar* ($\mathcal{G} =$ minor-excl(K_3, subdivision of $K_{1,3}$)), or a *forest* ($\mathcal{G} = \mathcal{T}_1$), or *outerplanar* ($\mathcal{G} =$ minor-excl$(K_4, K_{2,3})$), or *series-parallel*, or has treewidth ≤ 2 ($\mathcal{G} = \mathcal{T}_2$).

We consider the cases where $\mathcal{G} = \mathcal{T}_0$ and $\mathcal{G} = \mathcal{T}_1$ in the next two subsections.

3.4.1 Feedback Vertex Set

A *feedback vertex set (FVS)* of a graph G is a set U of vertices such that every cycle of G passes through at least one vertex of U. The previous known fixed-parameter algorithms for solving the k-feedback vertex set problem had running time $O((2k+1)^k n^2)$ [83] and alternatively time $O((917k^4)!(n+m))$ [30] (m is the number of edges.) Also there exists a randomized algorithm which needs $O(c4^k kn)$ time with probability at least $1-(1-\frac{1}{4^k})^{c4^k}$ [27]. The *k-feedback vertex set* problem ($\mathcal{FVS}$) asks whether an input graph has a feedback vertex set of size $\leq k$.

Kloks et al. [123] proved that the feedback vertex set problem on planar graphs of treewidth at most w can be solved in $O(c_{\mathsf{fvs}}^w n)$ time for some constant c_{fbs}. The complexity of their algorithm is based on the fact that the number of edges of a planar graph is bounded by a simple linear function of its vertices (i.e. $3n-6$). As we have similar bound $3n-5$ for $K_{3,3}(K_5)$-minor-free graphs [20, 118], one can easily observe that the algorithm of [123] works also for the more general case. Therefore, Theorem 3.20 can be applied for $\mathcal{G} = \mathcal{T}_1$ and $\delta = c_{\mathsf{fvs}}$ and we have the following.

Theorem 3.21. *If $\mathcal{G}$ is a graph class excluding some single-crossing graph H as a minor then*
(1) If G has a feedback vertex set of size at most k then G has treewidth at most $\max\{9.55\sqrt{k}, c_H\}$.
(2) We can check whether some n-vertex graph in $\mathcal{G}$ has a feedback vertex set of size $\leq k$ in $O(c_{\mathsf{fvs}}^{9.55\sqrt{k}}n + n^4)$ time.

Theorem 3.21 generalizes the results of [123] to any class of graphs excluding some single-crossing graph H as a minor.

3.4.2 Improving Bounds for Vertex Cover

Alber et al. [6] proved that planar graphs in $\mathcal{VC}_k$ have treewidth at most $4\sqrt{3}\sqrt{k}+5 < 6.93\sqrt{k}+5$. An easy improvement of this result is the folowing:

Lemma 3.22. *If a planar graph has a vertex cover of size $\leq k$ then its treewidth is bounded by* $5.52\sqrt{k}$

Proof. Again using Lemma 2.4, we may assume that G is a biconnected graph. Let S be a vertex cover in G where $|S| \leq k$. Consider a planar embedding of G.

Construct a triangulation H of G as follows: for any face F we add edges connecting only vertices of $F \cap S$. This operation constructs a triangulation as there is no pair of vertices in $F - S$ that are consecutive in F. Moreover, as all the added edges have endpoints in S, S is a vertex cover of H. We will prove that $\mathbf{tw}(H) \leq 5.52\sqrt{k}$.

We may assume that H is a triangulation without double edges. To see this, consider two edges e_1 and e_2 connecting vertices x and y and apply Lemma 2.4 on the graphs G_{in} and G_{ex} induced by the vertices included in each of the closed disks bounded by the cycle where the two edges of this cycle are identified.

Notice now that for each vertex $v \in V(H) - S$, all its neighbors are members of S. This means that $|V(H) - S| \leq r$ where r is the number of faces of $J = H[S]$. As H has not double edges, neither J has and therefore $|E(J)| \leq 3|V(J)| - 6$. It is known that $r \leq |E(J)| - |V(J)| + 2$ and we get that $r \leq 2|V(J)| - 4$. We conclude that $|V(H)| \leq |V(J)| + 2|V(J)| - 4 = 3|S| - 4 = 3k - 4$. From [92], we know that any n-vertex planar graph has treewidth at most $\frac{9}{2\sqrt{2}}\sqrt{n}$. This means that $\mathbf{tw}(H) \leq \frac{9}{2\sqrt{2}}\sqrt{3k-4} \leq \frac{9}{2\sqrt{2}}\sqrt{3}\sqrt{k}$. As G is a subgraph of H and $\frac{9}{2\sqrt{2}}\sqrt{3} < 5.52$, the result follows. □

Applying Theorem 3.20, we have that Condition (1) of Theorem 3.4 holds if $\mathcal{F}$ is the class of graphs with vertex cover $\leq k$ and $\mathcal{G}$ is any graph class excluding some single-crossing graph H as a minor when $c = c_H$, $\alpha_1 = 4$, $\alpha_2 = 4$, $\beta_1 = 5.52$, and $\beta_2 = 0$. Also, as we mentioned in Subsection 3.3.2 it is possible to decide in $O(2^w n)$ time if a graph has a vertex cover of size at most k. Therefore, Condition (2) holds for $\delta = 2$. Concluding, we have the following improvement of the results of Subsection 3.3.2 for any graph class excluding some single-crossing graph H as a minor.

Theorem 3.23. *If $\mathcal{G}$ is some graph class excluding some single-crossing graph H as a minor then the following hold.*
(1) If $G \in \mathcal{G}$ has a vertex cover of size at most k then G has treewidth at most $\max\{5.52\sqrt{k}, c_H\}$.
(2) There is an algorithm which checks whether some graph $G \in \mathcal{G}$ has a vertex cover of size $\leq k$ in $O(2^{5.52\sqrt{k}}k + kn + k^3 + n^4)$ time.

Because $\mathcal{EDS} \preccurlyeq_2 \mathcal{VC}$, we can also obtain an $O({c_{\mathsf{eds}}}^{5.52\sqrt{2k}}n + n^4)$-time algorithm for the edge dominating set problem on graphs excluding some single-crossing graph as a minor.

3.5 Further Extensions

In this section, we obtain fixed-parameter algorithms with exponential speedup for k-vertex cover and k-edge dominating set on classes of graphs that are not necessarily classifiable as single-crossing minor-free graphs. Our approach, similar to the Alber et al.'s approach [6], is a general one that can be applied to other problems.

Here we generalize the concept of *layerwise separation*, introduced by Alber's et al. [6] for planar graphs, to general graphs.

Definition 3.24. *Let G be a graph layered from a vertex v, and r be the number of layers. A* layerwise separation of width w and size s *for G is a sequence $(S_1, S_2, \cdots, S_r)$ of subsets of V, with property that $S_i \subseteq \bigcup_{j=i}^{i+(w-1)} L_j$; S_i separates layers L_{i-1} and L_{i+w}; and $\sum_{j=1}^{r} |S_j| \leq s$. Here we let $S_i = \emptyset$ for all $i < 1$ and $i > r$.*

Now we relate the concept of layerwise separation to parameterized problems.

Definition 3.25. *A parameterized problem P has* Layerwise Separation Property (LSP) *of width w and size-factor d, if for each instance (G, k) of the problem P, graph G admits a layerwise separation of width w and size dk.*

For example, we can obtain constants $w = 2$ and $d = 2$ for the vertex cover problem. In fact, consider a k-vertex cover C on a graph G and set $S_i = (L_i \cup L_{i+1}) \cap C$. The S_i's form a layerwise separation. Similarly, we can get constants $w = 2$ and $d = 4$ for the edge dominating set problem (see Alber et al. [6] for further examples).

Lemma 3.26. *Let P be a parameterized problem on instance (G, k) that admits a problem kernel of size dk. Then the parameterized problem P on the problem kernel has LSP of width 1 and size-factor d.*

Proof. Consider the problem kernel (G', k') for an instance (G, k) and obtain layering L' for G' from arbitrary vertex v. Then clearly the sequence $S_i = L'_i$ for $i = 1, \cdots, r'$ (r' is the number of layers), is a layerwise separation of width 1 and size $k' \leq dk$ for G'. □

In fact, using Lemma 3.26 and the problem kernel of size $2k$ (see Subsection 3.3.2) for the vertex cover problem, this problem has the LSP of width 1 and size-factor 2.

Now we are ready to present the main theorem of this section.

Theorem 3.27. *Suppose for a graph G from a minor-closed class of graphs, $\mathbf{ltw}(G) \leq cr + c'$ and a tree decomposition of width $ch + c'$ can be constructed in time $f(n, h)$ for any h consecutive layers. Also assume G admits a layerwise separation of width w and size dk. Then we have $\mathbf{tw}(G) \leq 2\sqrt{2cdk} + cw + c'$. Such a tree decomposition can be computed in time $O(kf(n, \sqrt{k}))$.*

Proof. The proof is very similar to the proof of Theorem 15 of Alber et al.'s work [6] and for the sake of brevity we only mention the differences and omit the lengthy details. In the proof, the concept of the kth outer face in planar graphs will be replaced by the concept of the kth layer (or level) in graphs of locally bounded treewidth. More precisely, Alber et al. [6] use the fact that treewidth of an h-outerplnar graph is $3h-1$, but in our proof we use the fact that, for any graph G, treewidth of any h consecutive layers is at most $ch + c'$ [103, 72]. In addition, as mentioned before, Eppstein [87] showed that a minor-closed graph class $\mathcal{E}$ has bounded local treewidth if and only if $\mathcal{E}$ is H-minor-free for some apex graph H. (A simpler proof of this theorem can be found in [65].) By Thomason [161], we know that any graph G excluding an r-clique as a minor cannot have more than $(0.319 + o(1))(r\sqrt{\log r})|V(G)|$ edges. This implies that for graph G mentioned in the statement of the theorem, $|E(G)| = O(|V(G)|)$, similar to the corresponding relation for planar graphs. This fact is used for analyzing the running time. □

Corollary 3.28. *For any H-minor-free graph G, where H is a single-crossing graph, that admits a layerwise separation of width w and size dk, we have $\mathbf{tw}(G) \leq 2\sqrt{6dk} + 3w + c_H$. Such a tree decomposition can be computed in time $O(k^{5/2}n + kn^4)$. Furthermore, for any $K_{3,3}(K_5)$-minor-free graph G, that admits a layerwise separation of width w and size dk, we have $\mathbf{tw}(G) \leq 2\sqrt{6dk} + 3w + 4$. Such a tree decomposition can be computed in time $O(k^{5/2}n)$ $(O(k^{5/2}n + kn^2))$.*

Proof. The proof follows directly from Theorem 3.27 and the fact that for any single-crossing-minor-free graph G, we can construct a tree decomposition of width $3h + c_h$ for any h consecutive layers in $O(h^3 \cdot n + n^4)$ time; for a $K_{3,3}$-minor-free or K_5-minor-free graph G, the running time can be reduced to $O(h^3 n)$ or $O(h^3 n + n^2)$, respectively [109, 72]. ($c_H = 4$ for these graphs.) $\square$

Finally, we have this general theorem.

Theorem 3.29. *Suppose for a graph G from a minor-closed class of graphs, $\mathbf{ltw}(G) \leq cr + c'$. Let P be a parameterized problem on G such that P has the LSP of width w and size-factor d and there exists an $O(\delta^w n)$-time algorithm, given a tree decomposition of width w for G, decides whether problem P has a solution of size k on G.*

Then there exists an algorithm which decides whether P has a solution of size k on G in time $O(2^{(11/3)(2\sqrt{2cdk}+cw+c')}n^{3.01} + \delta^{3.698(2\sqrt{2cdk}+cw+c')}n)$.

Proof. The proof follows from Theorem 3.27, the fact that for graph G, treewidth of any h consecutive layers is at most $ch + c'$ [103, 72], and finally the result of Amir [11], which says for any graph G of treewidth w, we can construct a tree decomposition of width at most $(11/3)w$ in time $O(2^{3.698w} n^{3.01})$. $\square$

For example, Theorem 3.29 gives an exponential speed up, i.e., an algorithm with running time $O(2^{O(\sqrt{gk})} k^{3.01} + kn + k^3 + n^4)$ (because $c = O(g)$ [87]), for solving vertex cover on graphs of bounded genus.

Recently, it was established that all minor-closed classes of graphs with bounded local treewidth, i.e., all minor-closed graph classes excluding an apex graph, in fact have linear local treewidth [66]. Therefore Theorem 3.29 applies generally to any such class of graphs.

3.6 Concluding Remarks

In this chapter, we considered H-minor-free graphs, where H is a single-crossing graph, and proved that if these graphs have a k-dominating set then their treewidth is at most $c\sqrt{k}$ for a small constant c. As a consequence, we obtained exponential speedup in designing FPT algorithms for several NP-hard problems on these graphs, especially $K_{3,3}$-minor-free or K_5-minor-free graphs. In fact, our approach is a general one that can be applied to several problems which can be reduced to the dominating set problem as discussed in Section 3.3 or to problems that themselves can be solved exponentially faster on planar graphs [6]. Here, we present several open problems that are possible extensions of results of this chapter.

One topic of interest is finding other problems to which the technique of this chapter can be applied. Moreover, it would be interesting to find other classes of graphs than H-minor-free graphs, where H is a single-crossing graph, on which the problems can be solved exponentially faster for parameter k. A partial answer to this question is the class of map graphs (see Chapter 4).

For several problems in this chapter, Kloks et al. [50, 123, 107, 122] introduced a reduction to the problem kernel on planar graphs. Because graphs excluding a single-crossing graph are similar to planar graphs, in the sense of having a linear number of edges and not having a clique of more than a constant size, we believe that one might obtain similar results for these graphs.

As mentioned before, Theorem 3.20 holds for any class of graphs with treewidth ≤ 2. It is an open problem whether it is possible to generalize it to apply to any class of graphs of treewidth $\leq h$ for arbitrary fixed h. Moreover, there exists no general method for designing $O(\delta^w n)$-time algorithms for vertex-removal problems in graphs with treewidth $\leq w$. If this becomes possible, then Theorem 3.23 will have considerable algorithmic applications.

Finally, as a matter of practical importance, it would be interesting to obtain a constant coefficient better than 9.55 for the treewidth of planar graphs having a k-dominating set, which would lead to a direct improvement on our results.

Chapter 4

Fixed-Parameter Algorithms for the (k, r)-Center in Planar Graphs and Map Graphs

Clustering is a key tool for solving a variety of application problems such as data mining, data compression, pattern recognition and classification, learning, and facility location. Among the algorithmic problem formulations of clustering are k-means, k-medians, and k-center. In all of these problems, the goal is to partition n given points into k *clusters* so that some objective function is minimized.

In this chapter, we concentrate on the (unweighted) (k, r)-*center* problem [26], in which the goal is to choose k *centers* from the given set of n points so that every point is within distance r from some center in the graph. In particular, the k-*center problem* [102] of minimizing the maximum distance to a center is exactly (k, r)-center when the goal is to minimize r subject to finding a feasible solution. In addition, the r-*domination problem* [26, 100] of choosing the fewest vertices whose r-neighborhoods cover the whole graph is exactly (k, r)-center when the goal is to minimize k subject to finding a feasible solution.

A sample application of the (k, r)-center problem in the context of facility location is the installation of emergency service facilities such as fire stations. Here we suppose that we can afford to buy k fire stations to cover a city, and we require every building to be within r city blocks from the nearest fire station to ensure a reasonable response time. Given an algorithm for (k, r)-center, we can vary k and r to find the best bicriterion solution according to the needs of the application. In this scenario, we can afford high running time (e.g., several weeks of real time) if the resulting solution builds fewer fire stations (which are extremely expensive) or has faster response time; thus, we prefer fixed-parameter algorithms over approximation algorithms.

In this application, and many others, the graph is typically planar or nearly so. Chen, Grigni, and Papadimitriou [58] have introduced a generalized notion of planarity which allows local nonplanarity. In this generalization, two countries of a map are adjacent if they share at least one point, and the resulting graph of adjacencies is called a *map graph*. (See Section 4.1 for a precise definition.) Planar graphs are the special case of map graphs in which at most three countries intersect at a point.

Previous results. r-domination and k-center are NP-hard even for planar graphs. For r-domination, the current best approximation (for general graphs) is a $(\log n+1)$-factor by phrasing the problem as an instance of set cover [26]. For k-center, there is a 2-approximation algorithm [102] which applies more generally to the case of weighted graphs satisfying the triangle inequality. Furthermore, no $(2 - \varepsilon)$-approximation algorithm exists for any $\varepsilon > 0$ even for unweighted planar graphs of maximum degree 3 [137]. For geometric k-center in which the weights are given by Euclidean distance in d-dimensional space, there is a PTAS whose running time is exponential in k [1]. Several relations between small r-domination sets for planar graphs and problems about organizing routing schemes with compact structures is discussed in [100].

The (k, r)-center problem can be considered as a generalization of the well-known dominating set problem. During the last two years in particular much attention has been paid to constructing fixed-parameter algorithms with exponential speedup for this problem. Alber et al. [2] were the first who demonstrated an algorithm checking whether a planar graph has a dominating set of size $\leq k$ in time $O(2^{70\sqrt{k}}n)$. This result was the first non-trivial result for the parameterized version of an NP-hard problem in which the exponent of the exponential term grows sublinearly in the parameter. Recently, the running time of this algorithm was further improved to $O(2^{27\sqrt{k}}n)$ [116]

and $O(2^{15.13\sqrt{k}}k + n^3 + k^4)$ [93]. Fixed-parameter algorithms for solving many different problems such as vertex cover, feedback vertex set, maximal clique transversal, and edge-dominating set on planar and related graphs such as single-crossing-minor-free graphs are considered in [74, 123](see also Chapter 3). Most of these problems have reductions to the dominating set problem. Also, because all these problems are closed under taking minors or contractions, all classes of graphs considered so far are minor-closed.

Our results. In this chapter, we focus on applying the tools of *parameterized complexity*, introduced by Downey and Fellows [83], to the (k, r)-center problem in planar and map graphs. We view both k and r as parameters to the problem. We introduce a new proof technique which allows us to extend known results on planar dominating set in two different aspects.

First, we extend the exponential speed-up for a generalization of dominating set, namely the (k, r)-center problem, on planar graphs. Specifically, the running time of our algorithm is $O((2r + 1)^{6(2r+1)\sqrt{k}+12r+3/2}n + n^4)$, where n is the number of vertices. Our proof technique is based on combinatorial bounds (Section 4.2) derived from the Robertson, Seymour, and Thomas theorem about quickly excluding planar graphs, and on a complicated dynamic program on graphs of bounded branchwidth (Section 4.3). Second, we extend our fixed-parameter algorithm to map graphs which is a class of graphs that is not minor-closed. In particular, the running time of the corresponding algorithm is $O((2r + 1)^{6(4r+1)\sqrt{k}+24r+3}n + n^4)$.

Notice that the exponential component of the running times of our algorithms depends only on the parameters, and is multiplicatively separated from the problem size n. Moreover, the contribution of k in the exponential part is sublinear. In particular, our algorithms have polynomial running time if $k = O(\log^2 n)$ and $r = O(1)$, or if $r = O(\log n/\log\log n)$ and $k = O(1)$. We stress the fact that we design our dynamic-programming algorithms using branchwidth instead of treewidth because this provides better running times.

Finally, in Section 4.5, we present several extensions of our results, including a PTAS for the r-dominating set problem and a generalization to a broad class of graph parameters.

4.1 Preliminary Results

In this section, we recall some definitions from Chapter 1 for which we present several preliminary results in more detail.

(k, r)-center. We say a graph G has a (k, r)-*center* or interchangeably has an r-*dominating set of size* k if there exists a set S of centers (vertices) of size at most k such that $N_G^r(S) = V(G)$. We denote by $\gamma_r(G)$ the smallest k for which there exists a (k, r)-center in the graph. One can easily observe that for any r the problem of checking whether an input graph has a (k, r)-center, parameterized by k is $W[2]$-hard by a reduction from dominating set. (See Downey and Fellows [83] for the definition of the W Hierarchy.)

Map graphs. Let Σ be a sphere. A Σ-*plane* graph G is a planar graph G drawn in Σ. To simplify notation, we usually do not distinguish between a vertex of the graph and the point of Σ used in the drawing to represent the vertex, or between an edge and the open line segment representing it. We denote the set of regions (faces) in the drawing of G by $R(G)$. (Every region is an open set.) An edge e or a vertex v is *incident* to a region r if $e \subseteq \bar{r}$ or $v \subseteq \bar{r}$, respectively. ($\bar{r}$ denotes the *closure* of r.)

For a Σ-plane graph G, a *map* $\mathcal{M}$ is a pair (G, ϕ), where $\phi : R(G) \to \{0, 1\}$ is a two-coloring of the regions. A region $r \in R(G)$ is called a *nation* if $\phi(r) = 1$ and a *lake* otherwise.

Let $N(\mathcal{M})$ be the set of nations of a map $\mathcal{M}$. The graph F is defined on the vertex set $N(\mathcal{M})$, in which two vertices r_1, r_2 are adjacent precisely if $\bar{r}_1 \cap \bar{r}_2$ contains at least one edge of G. Because F is the subgraph of the dual graph G^* of G, it is planar. Chen, Grigni, and Papadimitriou [58] defined the following generalization of planar graphs. A *map graph* $G_{\mathcal{M}}$ *of a map* $\mathcal{M}$ is the graph on the vertex set $N(\mathcal{M})$ in which two vertices r_1, r_2 are adjacent in $G_{\mathcal{M}}$ precisely if $\bar{r}_1 \cap \bar{r}_2$ contains at least one vertex of G.

Recall from Chapter 1 that we denote by G^k the *kth power of* G, i.e., the graph on the vertex set $V(G)$ such that two vertices in G^k are adjacent precisely if the distance in G between these vertices is at most k. Let G be a bipartite graph with a bipartition $U \cup W = V(G)$. The *half square* $G^2[U]$ is the graph on the vertex set U and two vertices are adjacent in $G^2[U]$ precisely if the distance between these vertices in G is 2.

Theorem 4.1 ([58]). *A graph $G_{\mathcal{M}}$ is a map graph if and only if it is the half-square of some planar bipartite graph H.*

Here the graph H is called a *witness for $G_{\mathcal{M}}$*. Thus the question of finding a (k,r)-center in a map graph $G_{\mathcal{M}}$ is equivalent to finding in a witness H of $G_{\mathcal{M}}$ a set $S \subseteq V(G_{\mathcal{M}})$ of size k such that every vertex in $V(G_{\mathcal{M}}) - S$ has distance $\leq 2r$ in H from some vertex of S.

The proof of Theorem 4.1 is constructive, i.e., given a map graph $G_{\mathcal{M}}$ together with its map $\mathcal{M} = (G, \phi)$, one can construct a witness H for $G_{\mathcal{M}}$ in time $O(|V(G_{\mathcal{M}})| + |E(G_{\mathcal{M}})|)$. One color class $V(G_{\mathcal{M}})$ of the bipartite graph H corresponds to the set of nations of the map $\mathcal{M}$. Each vertex v of the second color class $V(H) - V(G_{\mathcal{M}})$ corresponds to an intersection point of boundaries of some nations, and v is adjacent (in H) to the vertices corresponding to the nations it belongs. What is important for our proofs are the facts that

1. in such a witness, every vertex of $V(H) - V(G_{\mathcal{M}})$ is adjacent to a vertex of $V(G_{\mathcal{M}})$, and
2. $|V(H)| = O(|V(G_{\mathcal{M}})| + |E(G_{\mathcal{M}})|)$.

Thorup [162] provided a polynomial-time algorithm for constructing a map of a given map graph in polynomial time. However, in Thorup's algorithm, the exponent in the polynomial time bound is about 120 [57]. So from practical point of view there is a big difference whether we are given a map in addition to the corresponding map graph. Below we suppose that we are always given the map.

Branchwidth (see Section 2.4 for the definition and its relation to treewidth) is our main tool in this chapter. All our proofs can be rewritten in terms of the related and better-known parameter treewidth, and indeed treewidth would be easier to handle in our dynamic program. However, branchwidth provides better combinatorial bounds resulting in faster exponential speed-up of our algorithms.

The following deep result of Robertson, Seymour, and Thomas (Theorems (4.3) in [145] and (6.3) in [151]) plays an important role in the results of this thesis.

Theorem 4.2 ([151]). *Let $k \geq 1$ be an integer. Every planar graph with no $(k \times k)$-grid as a minor has branchwidth $\leq 4k - 3$.*

4.2 Combinatorial Bounds

Lemma 4.3. *Let $\rho, k, r \geq 1$ be integers and G be a planar graph having a (k, r)-center and with a $(\rho \times \rho)$-grid as a minor. Then $k \geq (\frac{\rho-2r}{2r+1})^2$.*

Proof. We set $V = \{1, \ldots, \rho\} \times \{1, \ldots, \rho\}$. Let

$$F = (V, \{((x, y), (x', y')) \mid |x - x'| + |y - y'| = 1\})$$

be a plane $(\rho \times \rho)$-grid that is a minor of some plane embedding of G. W.l.o.g. we assume that the external (infinite) face of this embedding of F is the one that is incident to the vertices of the set $V_{\text{ext}} = \{(x, y) \mid x = 1 \text{ or } x = \rho \text{ or } y = 1 \text{ or } y = \rho\}$, i.e., the vertices of F with degree < 4. We call the rest of the faces of F *internal faces*. We set $V_{\text{int}} = \{(x, y) \mid r + 1 \leq x \leq \rho - r,\ r + 1 \leq y \leq \rho - r\}$, i.e., V_{int} is the set of all vertices of F within distance $\geq r$ from all vertices in V_{ext}. Notice that $F[V_{\text{int}}]$ is a sub-grid of F and $|V_{\text{int}}| = (\rho - 2r)^2$. Given any pair of vertices $(x, y), (x', y') \in V$ we define $\delta((x, y), (x', y')) = \max\{|x - x'|, |y - y'|\}$.

We also define $d_F((x, y), (x', y'))$ to be the distance between any pair of vertices (x, y) and (x', y') in F. Finally we define J to be the graph occurring from F by adding in it the edges of the following sets:

$$\{((x, y), (x + 1, y + 1) \mid 1 \leq x \leq \rho - 1,\ 1 \leq y \leq \rho - 1)\}$$

$$\{((x, y+1), (x+1, y) \mid 1 \leq x \leq \rho - 1,\ 1 \leq y \leq \rho - 1)\}$$

(In other word we add all edges connecting pairs of non-adjacent vertices incident to its internal faces). It is easy to verify that $\forall (x,y),(x',y') \in V \ \ \delta((x,y),(x',y')) = d_J((x,y),(x',y'))$. This implies the following.

$$\text{If } R \text{ is a subgraph of } J, \text{ then } \forall (x,y),(x',y') \in V\ \delta((x,y),(x',y')) \leq d_R((x,y),(x',y')) \quad (4.1)$$

For any $(x,y) \in V$ we define $B_r((x,y)) = \{(a,b) \in V \mid \delta((x,y),(a,b)) \leq r\}$ and we observe the following:

$$\forall_{(x,y)\in V}\ |V(B_r((x,y)))| \leq (2r+1)^2. \tag{4.2}$$

Consider now the sequence of edge contractions/removals that transform G to F. If we apply on G only the contractions of this sequence we end up with a planar graph H that can obtained by the $(\rho \times \rho)$-grid F after adding edges to non-consecutive vertices of its faces. This makes it possible to partition the additional edges of H into two sets: a set denoted by E_1 whose edges connect non-adjacent vertices of some square face of F and another set E_2 whose edges connect pairs of vertices in V_{ext}. We denote by R the graph obtained by F if we add the edges of E_1 in F. As R is a subgraph of J, (4.1) implies that

$$\forall_{(x,y)\in V}\ N_R^r((x,y)) \subseteq B_r((x,y)) \tag{4.3}$$

We also claim that

$$\forall_{(x,y)\in V}\ N_H^r((x,y)) \subseteq B_r((x,y)) \cup (V - V_{\text{int}}) \tag{4.4}$$

To prove (4.4) we notice first that if we replace H by R in it then the resulting relation follows from (4.3). It remains to prove that the consecutive addition of edges of E_2 in R does not introduce in $N_R^r((x,y))$ any vertex of V_{int}. Indeed, this is correct because any vertex in V_{ext} is in distance $\geq r$ from any vertex in V_{int}. Notice now that (4.4) implies that $\forall_{(x,y)\in V}\ N_H^r((x,y)) \cap V_{\text{int}} \subseteq B_r((x,y)) \cap V_{\text{int}}$ and using (4.2) we conclude that

$$\forall_{(x,y)\in V}\ |N_H^r((x,y)) \cap V_{\text{int}}| \leq (2r+1)^2 \tag{4.5}$$

Let S be a (k', r)-center in the graph H. Applying (4.5) on S we have that the r-neighborhood of any vertex in S contains at most $(2r+1)^2$ vertices from V_{int}. Moreover, any vertex in V_{int} should belong to the r-neighborhood of some vertex in S. Thus $k' \geq |V_{\text{int}}|/(2r+1)^2 = (\rho - 2r)^2/(2r+1)^2$ and therefore $k' \geq (\frac{\rho-2r}{2r+1})^2$.

Clearly, the conditions that G has an r-dominating set of size k and $H \preceq_c G$ imply that H has an r-dominating set of size $k' \leq k$. (But this is not true for $H \preceq G$.) As H is a contraction of G and G has a (k, r)-center, we have that $k \geq k' \geq (\frac{\rho-2r}{2r+1})^2$ and lemma follows. □

We are ready to prove the main combinatorial result of this chapter:

Theorem 4.4. *For any planar graph G having a (k,r)-center,* $\mathbf{bw}(G) \leq 4(2r+1)\sqrt{k} + 8r + 1$.

Proof. Suppose that $\mathbf{bw}(G) > p = 4(2r+1)\sqrt{k} + 8r + \varepsilon - 3$ for some ε, $0 < \varepsilon \leq 4$, for which $p + 3 \equiv 0 \pmod 4$. By Theorem 4.2, G contains a $(\rho \times \rho)$-grid as a minor where $\rho = (2r+1)\sqrt{k} + 2r + \frac{\varepsilon}{4}$. By Lemma 4.3, $k \geq (\frac{\rho-2r}{2r+1})^2 = (\frac{(2r+1)\sqrt{k}+\frac{\varepsilon}{4}}{2r+1})^2$ which implies that $\sqrt{k} \geq \sqrt{k} + \frac{\varepsilon}{8r+4}$, a contradiction. □

Notice that the branchwidth of a map graph is unbounded in terms of k and r. For example, a clique of size n is a map graph and has a $(1,1)$-center and branchwidth $\geq 2/3n$.

Theorem 4.5. *For any map graph $G_{\mathcal{M}}$ having a (k,r)-center and its witness H,* $\mathbf{bw}(H) \leq 4(4r+3)\sqrt{k} + 16r + 9$.

Proof. The question of finding a (k,r)-center in a map graph $G_{\mathcal{M}}$ is equivalent to finding in a witness H of $G_{\mathcal{M}}$ a set $S \subseteq V(G_{\mathcal{M}})$ of size k such that every vertex $V(G_{\mathcal{M}}) - S$ is at distance $\leq 2r$ in H from some vertex of S. By the construction of the witness graph, every vertex of $V(H) - V(G_{\mathcal{M}})$ is adjacent to some vertex of $V(G_{\mathcal{M}})$. Thus H has a $(k, 2r+1)$-center and by Theorem 4.4 the proof follows. □

4.3 (k, r)-Centers in Graphs of Bounded Branchwidth

In this section, we present a dynamic-programming approach to solve the (k, r)-center problem on graphs of bounded branchwidth. It is easy to prove that, for a *fixed* r, the problem is in MSOL (monadic second-order logic) and thus can be solved in linear time on graphs of bounded treewidth (branchwidth). However, for r part of the input, the situation is more difficult. Additionally, we are interested in not just a linear-time algorithm but in an algorithm with running time $f(k, r)n$.

It is worth mentioning that our algorithm requires more than a simple extension of Alber et al.'s algorithm for dominating set in graphs of bounded treewidth [2], which corresponds to the case $r = 1$. In fact, finding a (k, r)-center is similar to finding homomorphic subgraphs, which has been solved only for special classes of graphs and even then only via complicated dynamic programs [105]. The main difficulty is that the path $v = v_0,\ v_1,\ v_2,\ \ldots,\ v_{\leq r} = c$ from a vertex v to its assigned center c may wander up and down the branch decomposition repeatedly, so that c and v may be in radically different 'cuts' induced by the branch decomposition. All we can guarantee is that the next vertex v_1 along the path from v to c is somewhere in a common 'cut' with v, and that vertex v_1 and v_2 are in a common 'cut', etc. In this way, we must propagate information through the v_i's about the remote location of c.

Let (T', τ) be a branch decomposition of a graph G with m edges and let $\omega' : E(T') \to 2^{V(G)}$ be the order function of (T', τ). We choose an edge $\{x, y\}$ in T', put a new vertex v of degree 2 on this edge, and make v adjacent to a new vertex r. By choosing r as a root in the new tree $T = T' \cup \{v, r\}$, we turn T into a rooted tree. For every edge of $f \in E(T) \cap E(T')$, we put $\omega(f) = \omega'(f)$. Also we put $\omega(\{x, v\}) = \omega(\{v, y\}) = \omega'(\{x, y\})$ and $\omega(\{r, v\}) = \emptyset$.

For an edge f of T we define E_f (V_f) as the set of edges (vertices) that are "below" f, i.e., the set of all edges (vertices) g such that every path containing g and $\{v, r\}$ in T contains f. With such a notation, $E(T) = E_{\{v,r\}}$ and $V(T) = V_{\{v,r\}}$. Every edge f of T that is not incident to a leaf has two children that are the edges of E_f incident to f. We denote by G_f the subgraph of G induced by the vertices incident to edges from the following set

$$\{\tau^{-1}(x) \mid x \in V_f \wedge (x \text{ is a leaf of } T')\}.$$

The subproblems in our dynamic program are defined by a coloring of the vertices in $\omega(f)$ for every edge f of T. Each vertex will be assigned one of $2r + 1$ colors

$$\{0, \uparrow 1, \uparrow 2, \ldots, \uparrow r, \downarrow 1, \downarrow 2, \ldots, \downarrow r\}.$$

The meaning of the color of a vertex v is as follows:

- 0 means that the vertex v is a chosen center.
- $\downarrow i$ means that vertex v has distance exactly i to the closest center c. Moreover, there is a neighbor $u \in V(G_f)$ of v that is at distance exactly $i - 1$ to the center c. We say that neighbor u *resolves* vertex v.
- $\uparrow i$ means that vertex v has distance exactly i to the closest center c. However, there is *no* neighbor of v in $V(G_f)$ resolving v. Thus we are guessing that any vertex resolving v is somewhere in $V(G) - V(G_f)$.

Intuitively, the vertices colored by $\downarrow i$ have already been resolved (though the vertex that resolves it may not itself be resolved), whereas the vertices colored by $\uparrow i$ still need to be assigned vertices that are closer to the center.

We use the notation $\updownarrow i$ to denote a color of either $\uparrow i$ or $\downarrow i$. Also we use $\updownarrow 0 = 0$.

For an edge f of T, a coloring of the vertices in $\omega(f)$ is called *locally valid* if the following property holds: for any two adjacent vertices v and w in $\omega(f)$, if v is colored $\updownarrow i$ and w is colored $\updownarrow j$, then $|i - j| \leq 1$. (If the distance from some vertex v to the closest center is i, then for every neighbor u of v the distance from u to the closest center can not be less than $i-1$ or more than $i+1$.)

For each locally valid coloring c of $\omega(f)$, $f \in E(T)$, we define $A_f(c)$ as the size of the "minimum (k,r)-center restricted to G_f and coloring c". More precisely, $A_f(c)$ is the minimum cardinality of a set $D_f(c) \subseteq V(G_f)$ such that

- For every vertex $v \in \omega(f)$,
  - $c(v) = 0$ if and only if $v \in D_f(c)$, and
  - if $c(v) = \downarrow i$, $i \geq 1$, then $v \notin D_f(c)$ and either there is a vertex $u \in \omega(f)$ colored by $\updownarrow j$, $j < i$, at distance $i-j$ from v in G_f, or there is a path P of length i in G_f connecting v with some vertex of $D_f(c)$ such that no inner vertex of P is in $\omega(f)$.
- Every vertex $v \in V(G_f) - \omega(f)$ whose closest center is at distance i, either is at distance i in G_f from some center in $D_f(c)$, or is at distance j, $j < i$, in G_f from a vertex $u \in \omega(f)$ colored $\updownarrow(i-j)$.

We put $A_f(c) = +\infty$ if there is no such a set $D_f(c)$, or if c is not a locally valid coloring. Because $\omega(\{r,v\}) = \emptyset$ and $G_{\{r,v\}} = G$, we have that $A_{\{r,v\}}(c)$ is the smallest size of an r-dominating set in G.

We start computations of the functions A_f from leaves of T. Let x be a leaf of T and let f be the edge of T incident with x. Then G_f is the edge $\{u,v\}$ of G corresponding to x and either $V(G_f) = \omega(f)$, or the vertex $u = V(G_f) - \omega(f)$ is the vertex of degree 1 in G. If $V(G_f) = \omega(f)$, we consider all locally valid colorings c of $\omega(f)$ such that if a vertex $v \in \omega(f)$ is colored by $\downarrow i$ for $i > 0$ then u is colored by $\updownarrow i-1$. Let us note that there is always an optimal solution containing no centers in vertices of degree 1. Thus in the case $v = V(G_f) - \omega(f)$ we color v in one of the colors from the set $\{0, \uparrow 1, \uparrow 2, \dots, \uparrow r-1\}$. For each such coloring c we define $A_f(c)$ to be the number of vertices colored 0 in $\omega(f)$. Otherwise, $A_f(c)$ is $+\infty$, meaning that this coloring c is infeasible. The brute-force algorithm takes $O(rm)$ time for this step.

Let f be a non-leaf edge of T and let f_1, f_2 be the children of f. Define $X_1 = \omega(f) - \omega(f_2)$, $X_2 = \omega(f) - \omega(f_1)$, $X_3 = \omega(f) \cap \omega(f_1) \cap \omega(f_2)$, and $X_4 = (\omega(f_1) \cup \omega(f_2)) - \omega(f)$.

Notice that

$$\omega(f) = X_1 \cup X_2 \cup X_3. \tag{4.6}$$

By the definition of ω, it is impossible that a vertex belongs to exactly one of $\omega(f), \omega(f_1), \omega(f_2)$. Therefore, condition $u \in X_4$ implies that $u \in \omega(f_1) \cap \omega(f_2)$ and we conclude that

$$\omega(f_1) = X_1 \cup X_3 \cup X_4, \tag{4.7}$$

and

$$\omega(f_2) = X_2 \cup X_3 \cup X_4. \tag{4.8}$$

We say that a coloring $c \in \{0, \uparrow 1, \uparrow 2, \ldots, \uparrow r, \downarrow 1, \downarrow 2, \ldots, \downarrow r\}^{|\omega(f)|}$ of $\omega(f)$ is *formed* from a coloring c_1 of $\omega(f_1)$ and a coloring c_2 of $\omega(f_2)$ if

1. For every $u \in X_1$, $c(u) = c_1(u)$;
2. For every $u \in X_2$, $c(u) = c_2(u)$;
3. For every $u \in X_3$,
   (a) If $c(u) = \uparrow i$, $1 \leq i \leq r$, then $c(u) = c_1(u) = c_2(u)$. Intuitively, because vertex u is unresolved in $\omega(f)$, this vertex is also unresolved in $\omega(f_1)$ and in $\omega(f_2)$.
   (b) If $c(u) = 0$ then $c_1(u) = c_2(u) = 0$.
   (c) If $c(u) = \downarrow i$, $1 \leq i \leq r$, then $c_1(u), c_2(u) \in \{\downarrow i, \uparrow i\}$ and $c_1(u) \neq c_2(u)$. We avoid the case when both c_1 and c_2 are colored by $\downarrow i$ because it is sufficient to have the vertex u resolved in at least one coloring. This observation helps to decrease the number of colorings forming a coloring c. (Similar

arguments using a so-called "monotonicity property" are made by Alber et al. [2] for computing the minimum dominating set on graphs of bounded treewidth.)

4. For every $u \in X_4$,

   (a) either $c_1(u) = c_2(u) = 0$ (in this case we say that u is *formed by* 0 *colors*),

   (b) or $c_1(u), c_2(u) \in \{\downarrow i, \uparrow i\}$ and $c_1(u) \neq c_2(u)$, $1 \leq i \leq r$ (in this case we say that u is *formed by* $\{\downarrow i, \uparrow i\}$ *colors*).

   This property says that every vertex u of $\omega(f_1)$ and $\omega(f_2)$ that does not appear in $\omega(f)$ (and hence does not appear further) should finally either be a center (if both colors of u in c_1 and c_2 were 0), or should be resolved by some vertex in $V(G_f)$ (if one of the colors $c_1(u), c_2(u)$ is $\downarrow i$ and one $\uparrow i$). Again, we avoid the case of $\downarrow i$ in both c_1 and c_2.

Notice that every coloring of $\omega(f)$ is formed from some colorings of $\omega(f_1)$ and $\omega(f_2)$. Moreover, if $D_f(c)$ is the restriction to G_f of some (k,r)-center and such a restriction corresponds to a coloring c of $\omega(f)$ then $D_f(c)$ is the union of the restrictions $D_{f_1}(c_1)$, $D_{f_2}(c_2)$ to G_{f_1}, G_{f_2} of two (k,r)-centers where these restrictions correspond to some colorings c_1, c_2 of $\omega(f_1)$ and $\omega(f_2)$ that form the coloring c.

We compute the values of the corresponding functions in a bottom-up fashion. The main observation here is that if f_1 and f_2 are the children of f, then the vertex sets $\omega(f_1)$ $\omega(f_2)$ "separate" subgraphs G_{f_1} and G_{f_2}, so the value $A_f(c)$ can be obtained from the information on colorings of $\omega(f_1)$ and $\omega(f_2)$.

More precisely, let c be a coloring of $\omega(f)$ formed by colorings c_1 and c_2 of f_1 and f_2. Let $\#_0(X_3, c)$ be the number of vertices in X_3 colored by color 0 in coloring c and and let $\#_0(X_4, c)$ be the number of vertices in X_4 formed by 0 colors. For a coloring c we assign

$$A_f(c) = \min\{A_{f_1}(c_1) + A_{f_2}(c_2) - \#_0(X_3, c_1) - \#_0(X_4, c_1) \mid c_1, c_2 \text{ form } c\}. \quad (4.9)$$

(Every 0 from X_3 and X_4 is counted in $A_{f_1}(c_1) + A_{f_2}(c_2)$ twice and $X_3 \cap X_4 = \emptyset$.)

The time to compute the minimum in (4.9) is given by

$$O\Big(\sum_c \Big|\Big\{\{c_1, c_2\} \mid c_1, c_2 \text{ form } c\Big\}\Big|\Big).$$

Let $x_i = |X_i|$, $1 \le i \le 4$. For a coloring c let z_3 be the number of vertices colored by $\downarrow$ colors in X_3. Also we denote by z_4 the number of vertices in X_4 formed by $\{\downarrow i, \uparrow i\}$ colors, $1 \le i \le r$. Thus the number of pairs forming c is $2^{z_3+z_4}$. The number of colorings of $\omega(f)$ such that exactly z_3 vertices of X_3 are colored by $\downarrow$ colors and such that exactly z_4 vertices of X_4 are formed by $\{\downarrow, \uparrow\}$ colors is

$$(2r+1)^{x_1}(2r+1)^{x_2}(r+1)^{x_3-z_3}\binom{x_3}{z_3} r^{z_3}\binom{x_4}{z_4} r^{z_4}.$$

Thus the number of operations needed to estimate (4.9) for all possible colorings of $\omega(f)$ is

$$\sum_{p=0}^{x_3}\sum_{q=0}^{x_4} 2^{p+q}(2r+1)^{x_1+x_2}(r+1)^{x_3-p}\binom{x_3}{p} r^p \binom{x_4}{q} r^q = (2r+1)^{x_1+x_2+x_4}(3r+1)^{x_3}.$$

Let ℓ be the branchwidth of G. The sets X_i, $1 \le i \le 4$, are pairwise disjoint and by (4.6)–(4.8),

$$\begin{aligned} x_1 + x_2 + x_3 &\le \ell \\ x_1 + x_3 + x_4 &\le \ell \\ x_2 + x_3 + x_4 &\le \ell. \end{aligned} \tag{4.10}$$

The maximum value of the linear function $\log_{3r+1}(2r+1)\cdot(x_1+x_2+x_4)+x_3$ subject to the constraints in (4.10) is $\frac{3\log_{3r+1}(2r+1)}{2}\ell$. (This is because the value of the corresponding LP achieves maximum at $x_1 = x_2 = x_4 = 0.5\ell$, $x_3 = 0$.) Thus one can evaluate (4.9) in time

$$(2r+1)^{x_1+x_2+x_4}(3r+1)^{x_3} \le (3r+1)^{\frac{3\log_{3r+1}(2r+1)}{2}\ell} = (2r+1)^{\frac{3}{2}\cdot\ell}.$$

It is easy to check that the number of edges in T is $O(m)$ and the time needed to evaluate $A_{\{r,v\}}(c)$ is $O((2r+1)^{\frac{3}{2}\cdot\ell}m)$. Moreover, it is easy to modify the algorithm to obtain an optimal choice of centers by bookkeeping the colorings assigned to each set $\omega(f)$.

Summarizing, we obtain the following theorem:

Theorem 4.6. *For a graph G on m edges and with a given branch decomposition of width $\leq \ell$, and integers k, r, the existence of a (k,r)-center in G can be checked in $O((2r+1)^{\frac{3}{2}\cdot\ell}m)$ time and, in case of a positive answer, constructs a (k,r)-center of G in the same time.*

Similar result can be obtained for map graphs.

Theorem 4.7. *Let H be a witness of a map graph $G_{\mathcal{M}}$ on n vertices and let k, r be integers. If a branch-decomposition of width $\leq \ell$ of H is given, the existence of a (k,r)-center in $G_{\mathcal{M}}$ can be checked in $O((2r+1)^{\frac{3}{2}\cdot\ell}n)$ time and, in case of a positive answer, constructs a (k,r)-center of G in the same time.*

Proof. We give a sketch of the proof here. H is bipartite graph with a bipartition $(V(G_{\mathcal{M}}), V(H) - V(G_{\mathcal{M}}))$. There is a (k,r)-center in $G_{\mathcal{M}}$ if and only if H has a set $S \subseteq V(G_{\mathcal{M}})$ of size k such that every vertex $V(G_{\mathcal{M}}) - S$ is at distance $\leq 2r$ in H from some vertex of S. We check whether such a set S exists in H by applying arguments similar the proof of Theorem 4.6. The main differences in the proof are the following. Now we color vertices of the graph H by $\updownarrow i$, $0 \leq i \leq 2r$, where i is even. Thus we are using $2r+1$ numbers. Because we are not interested whether the vertices of $V(H) - V(G_{\mathcal{M}})$ are dominated or not, for vertices of $V(H) - V(G_{\mathcal{M}})$ we keep the same number as for a vertex of $V(G_{\mathcal{M}})$ resolving this vertex. For a vertex in $V(G_{\mathcal{M}})$ we assign a number $\downarrow i$ if there is a resolving vertex from $V(H) - V(G_{\mathcal{M}})$ colored $\updownarrow (i-2)$. Also we change the definition of locally valid colorings: for any two adjacent vertices v and w in $\omega(f)$, if v is colored $\updownarrow i$ and w is colored $\updownarrow j$, then $|i - j| \leq 2$.

Finally, H is planar, so $|E(H)| = O(|V(H)|) = O(n)$. $\square$

4.4 Algorithms for the (k, r)-Center Problem

For a planar graph G and integers k, r, we solve (k, r)-center problem on planar graphs in three steps.

Step 1: We check whether the branchwidth of G is at most $4(2r+1)\sqrt{k}+8r+1$. This step requires $O((|V(G)|+|E(G)|)^2)$ time according to the algorithm due to Seymour & Thomas (algorithm (7.3) of Section 7 of [155] — for an implementation, see the results of Hicks [113]). If the answer is negative then we report that G has no any (k, r)-center and stop. (The correctness of this step is verified by Theorem 4.4.) Otherwise go to the next step.

Step 2: Compute an optimal branch-decomposition of a graph G. This can be done by the algorithm (9.1) in the Section 9 of [155] which requires $O((|V(G)|+|E(G)|)^4)$ steps.

Step 3: Compute, if it exists, a (k, r)-center of G using the dynamic-programming algorithm of Section 4.3.

It is crucial that, for practical applications, there are no *large hidden constants* in the running time of the algorithms in Steps 1 and 2 above. Because for planar graphs $|E(G)| = O(|V(G)|)$, we conclude with the following theorem:

Theorem 4.8. *There exists an algorithm finding, if it exists, a (k, r)-center of a planar graph in $O((2r+1)^{6(2r+1)\sqrt{k}+12r+3/2}n + n^4)$ time.*

Similar arguments can be applied to solve the (k, r)-center problem on map graphs. Let $G_{\mathcal{M}}$ be a map graph. To check whether $G_{\mathcal{M}}$ has a (k, r)-center, we compute optimal branchwidth of its witness H. By Theorem 4.5, if $\mathbf{bw}(H) > 4(4r+3)\sqrt{k} + 16r+9$, then $G_{\mathcal{M}}$ has no (k, r)-center. If $\mathbf{bw}(H) \leq 4(4r+3)\sqrt{k}+16r+9$, then by Theorem 4.7 we obtain the following result:

Theorem 4.9. *There exists an algorithm finding, if it exists, a (k, r)-center of a map graph in $O((2r+1)^{6(4r+1)\sqrt{k}+24r+13.5}n + n^4)$ time.*

By a straightforward modification to the dynamic program, we obtain the same results for the *vertex-weighted (k,r)-center* problem, in which the vertices have real weights and the goal is to find a (k,r)-center of minimum total weight.

4.5 Concluding Remarks

In this chapter, we presented fixed-parameter algorithms with exponential speed-up for the (k,r)-center problem on planar graphs and map graphs. Our methods for (k,r)-center can also be applied to algorithms on more general graph classes like constant powers of planar graphs, which are not minor-closed family of graphs. Extending these results to other non-minor-closed families of graphs would be instructive.

Surprisingly, the algorithm described in Section 4.4 does not use special properties of (k,r)-center problem at all. The only properties the algorithm really needs are the combinatorial bound used in Step 1 and the fact that the problem can be solved on graphs of bounded branchwidth (Step 3). The proof of combinatorial bound (Theorem 4.4) is based on excluded planar graphs theorem of Robertson, Seymour & Thomas and the following two facts used in the proof of Lemma 4.3.

Fact 1: (k,r)-center problem is closed under edge contraction operation

Fact 2: For any partially triangulated grid (a *partially triangulated $(\rho\times\rho)$-grid* is any graph obtained by adding non-crossing edges between pairs of nonconsecutive vertices on a common face of a planar embedding of an $(\rho\times\rho)$-grid) having a (k,r)-center, k is at least $(\frac{\rho-2r}{2r+1})^2$.

The above observations, summarized in the next theorem, provide a general and versatile approach for many parameterized problems.

Theorem 4.10. *Let* $\mathbf{p}$ *be a function mapping graphs to non-negative integers such that the following conditions are satisfied:*

(1) *There exists an algorithm checking whether* $\mathbf{p}(G)\le w$ *in* $f(\mathbf{bw}(G))n^{O(1)}$ *steps.*

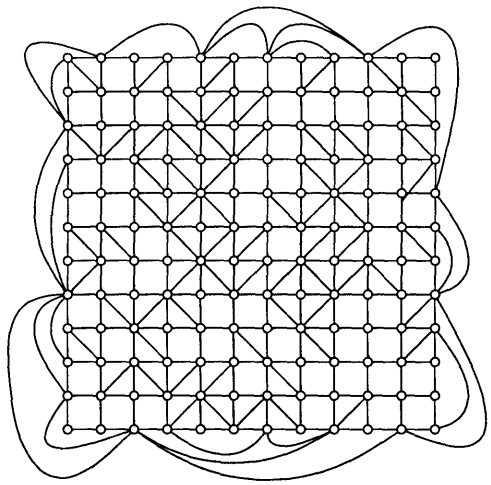

Figure 4-1: A partially triangulated (12×12)-grid.

(2) *For any $k \geq 0$, the class of graphs where $\mathbf{p}(G) \leq k$ is closed under taking of contractions.*

(3) *For any partially triangulated $(\rho \times \rho)$-grid R, $\mathbf{p}(R) = \Omega(\rho^2)$.*

Then there exists an algorithm checking whether $p(G) \leq k$ on planar graphs in $O(f(\sqrt{k}))n^{O(1)}$ steps.

For a wide source of parameters satisfying condition (1) we refer to the theory of Courcelle [61] (see also [15]). Apart from (k, r)-center and dominating set, examples of parameters satisfying conditions (2) and (3) are vertex cover, feedback vertex set, minimum maximal matching, edge dominating set and many others. For parameters where $f(\mathbf{bw}(G)) = 2^{O(\mathbf{bw}(G))}$, this result is a strong generalization of Alber et al.'s approach which requires that the problem of checking whether $\mathbf{p}(G) \leq k$ should satisfy the "layerwise separation property" [6]. Moreover, the algorithms involved are expected to have better constants in their exponential part comparatively to the ones appearing in [6]. More generally, it seems that our approach should extend other graph algorithms (not just dominating-set-type problems) to apply to the rth power and/or half-square of a graph (and hence in particular map graphs). It would be interesting to explore to which other problems our approach can be applied. Also, obtaining "fast" algorithms for problems like feedback vertex set or vertex cover on constant powers of graphs of bounded branchwidth (treewidth), as we did for dominating set, would be interesting.

In addition, there are several interesting variations on the (k, r)-center problem. In *multiplicity-m* (k, r)*-center*, the k centers must satisfy the additional constraint that every vertex is within distance r of at least m centers. In f*-fault-tolerant* (k, r)*-center* [26], every non-center vertex must have f vertex-disjoint paths of length at most r to centers. (For this problem with $r = \infty$, [26] gives a polynomial-time $O(f \log |V|)$-approximation algorithm for k.) In L*-capacitated* (k, r)*-center* [26], each of the k centers can satisfy only L "customers", essentially forcing the assignment of vertices to centers to be load-balanced. (For this problem, [26] gives a polynomial-time $O(\log |V|)$-approximation algorithm for r.) In *connected* (k, r)*-center* [157], the k chosen centers must form a connected subgraph. In all these problems, the main challenge is to design the dynamic program on graphs of bounded treewidth/branchwidth. We believe that our approach can be used as the main guideline in this direction.

Map graphs can be seen as contact graphs of disc homeomorphs. A question is whether our results can be extended for another geometric classes of graphs. An interesting candidate is the class of unit-disk graphs. The current best algorithms for finding a vertex cover or a dominating set of size k on these graphs have $n^{O(\sqrt{k})}$ running time [7].

Using our results we can also easily obtain a PTAS for r-dominating set on planar and map graphs. These results are similar to the approximation algorithms for independent set on map graphs by Chen [57]. We combine Theorems 4.6 and 4.7 with the approaches of Eppstein [87] and Grohe [103] (which in turn are based on the classic Baker's approach [23]), and adapt these approaches to branch decompositions instead of tree decompositions. We obtain a $(1 + 2r/p)$-approximation algorithm for r-domination in planar graphs with running time $O(p(2r + 1)^{3(p+2r)}m)$, and for map graphs we obtain a $(1 + 4r/p)$-approximation algorithm with running time $O(p(4r + 3)^{3(p+4r)}m)$.

Chapter 5

Subexponential Parameterized Algorithms on Bounded-Genus Graphs and H-Minor-Free Graphs

Algorithms for H-minor-free graphs for a fixed graph H have been studied extensively; see e.g. [51, 104, 53, 120, 138]. In particular, it is generally believed that several algorithms for planar graphs can be generalized to H-minor-free graphs for any fixed H [104, 120, 138]. H-minor-free graphs are very general. The deep Graph-Minor Theorem of Robertson and Seymour shows that any graph class that is closed under minors is characterized by excluding a finite set of minors. In particular, any graph class that is closed under minors (other than the class of all graphs) excludes at least one minor H.

In this chapter, we introduce a framework for extending algorithms for planar graphs to apply to H-minor-free graphs for any fixed H. In particular, we design subexponential fixed-parameter algorithms for dominating set, vertex cover, and set cover (viewed as one-sided domination in a bipartite graph) for H-minor-free graphs. Our framework consists of three components, as described below. We believe that many of these components can be applied to other problems and conjectures as well (see e.g., Chapter 8 for another application of this framework).

First we extend the algorithm for planar graphs to bounded-genus graphs. Roughly speaking, we study the structure of the solution to the problem in $k \times k$ grids, which form a representative substructure in both planar graphs and bounded-genus graphs, and capture the main difficulty of the problem for these graphs. Then using Robertson and Seymour's graph-minor theory, we repeatedly remove handles to reduce the bounded-genus graph down to a planar graph, which is essentially a grid.

Second we extend the algorithm to *almost-embeddable* graphs which can be drawn in a bounded-genus surface except for a bounded number of "local areas of non-planarity", called vortices, and for a bounded number of "apex" vertices, which can have any number of incident edges that are not properly embedded (see Section 1.3 for precise definitions). Because each vortex has bounded pathwidth, the number of vortices is bounded, and the number of apices is bounded, we are able to extend our approach to solve almost-embeddable graphs using our solution to bounded-genus graphs.

Third we apply a deep theorem of Robertson and Seymour which characterizes H-minor-free graphs as a tree structure of pieces, where each piece is an almost-embeddable graph. Using dynamic programming on such tree structures, analogous to algorithms for graphs of bounded treewidth, we are able to combine the pieces and solve the problem for H-minor-free graphs. Note that the standard bounded-treewidth methods do not suffice for general H-minor-free graphs, unlike e.g. bounded-genus graphs, because their treewidth can be arbitrarily large with respect to the parameter [62] (see Chapter 7). Our contribution is to overcome this barrier algorithmically using a two-level dynamic program in a more general tree structure called a "clique-sum decomposition".

The first step of this procedure, for bounded-genus graphs, applies to a broad class of problems called "bidimensional problems". Roughly speaking, a parameterized problem is *bidimensional* if the parameter is large (linear) in a grid and closed under contractions. Examples of bidimensional problems include vertex cover, feedback vertex set, minimum maximal matching, dominating set, edge dominating set, clique-transversal set, and set cover. We obtain subexponential fixed-parameter algorithms for all of these problems in bounded-genus graphs. As a special case, this generalization settles an open problem about dominating set posed by Ellis, Fan, and Fellows [84]. Along the way, we establish an upper bound on the treewidth (or branchwidth) of a bounded-genus graph that excludes some planar graph H as a minor (See Section 2.4 for the exact definition of branchwidth and its relation to treewidth). This bound depends linearly on the size $|V(H)|$ of the excluded graph H and the genus $g(G)$ of the graph G, and applies and extends the graph-minors work of Robertson and Seymour.

This chapter is organized as follows. Section 5.1 is devoted to graphs on surfaces. We construct a general framework for obtaining subexponential parameterized algorithms on graphs of bounded genus. First we introduce the concept of bidimensional problem, and then prove that every bidimensional problem has a subexponential parameterized algorithm on graphs of bounded genus. The proof techniques used in this section are very indirect and are based on deep theorems from Robertson and Seymour's Graph Minors XI and XII. As a byproduct of our results we obtain a generalization of Quickly Excluding a Planar Graph Theorem for graphs of bounded genus. In Section 5.3 we make a further step by developing subexponential algorithms for graphs containing no fixed graph H as a minor. The proof of this result is based on combinatorial bounds from the previous section, a deep structural theorem from Graph Minors XIV, and complicated dynamic programming. Finally, in Section 5.4, we present several extensions of our results and some open problems.

5.1 Graphs on Surfaces

5.1.1 Preliminaries

In this subsection we describe some of the machinery developed in the Graph Minors series that we use in our proofs.

A *surface* Σ is a compact 2-manifold without boundary. A *line* in Σ is a subset homeomorphic to $[0,1]$. An O-arc is a subset of Σ homeomorphic to a circle. A subset of Σ is an *open disc* if it is homeomorphic to $\{(x,y) \mid x^2+y^2<1\}$, and it is a *closed* *disc* if it is homeomorphic to $\{(x,y) \mid x^2+y^2 \le 1\}$.

A *2-cell embedding* of a graph G in a surface Σ is a drawing of the vertices as points in Σ and the edges as lines in Σ such that every region (face) bounded by edges is an open disc. To simplify notation, we do not distinguish between a vertex of G and the point of Σ used in the drawing to represent the vertex, or between an edge and the line representing it. We also consider G as the union of points corresponding to its vertices and edges. Also, a subgraph H of G can be seen as a graph H where $H \subseteq G$. A *region* of G is a connected component of $\Sigma - E(G) - V(G)$. (Every region is an open disc.) We use the notation $V(G)$, $E(G)$, and $R(G)$ for the set of the vertices, edges, and regions of G.

If $\Delta \subseteq \Sigma$, then $\overline{\Delta}$ denotes the *closure* of Δ, and the boundary of Δ is $\mathbf{bd}(\Delta) = \overline{\Delta} \cap \overline{\Sigma - \Delta}$. A vertex or an edge x is *incident* to a region r if $x \subseteq \mathbf{bd}(r)$.

A subset of Σ meeting the drawing only at vertices of G is called *G-normal.* If an O-arc is G-normal, then we call it a *noose*. The length of a noose is the number of its vertices. We say that a disc D is *bounded* by a noose N if $N = \mathbf{bd}(D)$. A graph G 2-cell embedded in a connected surface Σ is *θ-representative* if every noose of length less than θ is contractable (null-homotopic in Σ).

Tangles were introduced by Robertson and Seymour in [145]. A *separation* of a graph G is a pair (A,B) of subgraphs with $A \cup B = G$ and $E(A \cap B) = \emptyset$, and its *order* is $|V(A \cap B)|$. A *tangle of order* $\theta \ge 1$ is a set $\mathcal{T}$ of separations of G, each of order less than θ, such that

1. for every separation (A,B) of G of order less than θ, $\mathcal{T}$ contains one of (A,B) and (B,A);

2. if $(A_1,B_1),(A_2,B_2),(A_3,B_3) \in \mathcal{T}$, then $A_1 \cup A_2 \cup A_3 \neq G$; and

3. if $(A,B) \in \mathcal{T}$, then $V(A) \neq V(G)$.

Let G be a graph 2-cell embedded in a connected surface Σ. A tangle $\mathcal{T}$ of order θ is *respectful* if, for every noose N in Σ of length less than θ, there is a closed disc $\Delta \subseteq \Sigma$ with $\mathbf{bd}(\Delta) = N$ such that the separation $(G \cap \Delta, G \cap \overline{\Sigma - \Delta}) \in \mathcal{T}$.

Our proofs are based on the following results from the Graph Minors series of papers by Robertson and Seymour.

Theorem 5.1. [145, (4.3)] *Let G be a graph with at least one edge. Then there is a tangle in G of order θ if and only if G has branchwidth at least θ.*

Theorem 5.2. [146, (4.1)] *Let Σ be a connected surface, not homeomorphic to a sphere; let $\theta \geq 1$; and let G be a θ-representative graph 2-cell embedded in Σ. Then there is a unique respectful tangle in G of order θ.*

Our proofs also use the notion of the radial graph. Informally, the radial graph of a graph G 2-cell embedded in Σ is the bipartite graph R_G obtained by selecting a point in every region r of G and connecting it via an edge to every vertex of G incident to that region. However, a region may be incident to the same vertex "more than once", so we need a more formal definition. Precisely, R_G is a *radial graph* of a graph G 2-cell embedded in Σ if

1. $E(G) \cap E(R_G) = V(G) \subseteq V(R_G)$;
2. each region $r \in R(G)$ contains a unique vertex $v_r \in V(R_G)$;
3. R_G is bipartite with a bipartition $(V(G), \{v_r \colon r \in R(G)\})$;
4. if e, f are edges of R_G with the same ends $v \in V(G)$, $v_r \in V(R_G)$, then $e \cup f$ does not bound a closed disc in $r \cup \{v\}$; and
5. R_G is maximal subject to Conditions 1–4.

5.1.2 Bounding the Representativity

Define the *$(r \times r)$-grid* to be the graph on r^2 vertices $\{(x, y) \mid 1 \leq x, y \leq r\}$ with edges between vertices differing by ± 1 in exactly one coordinate. A *partially triangulated $(r \times r)$-grid* is any planar supergraph of the $(r \times r)$-grid.

Lemma 5.3. *Let G be a graph 2-cell embedded in a surface Σ, not homeomorphic to a sphere, of representativity at least θ. Then G contains as a contraction a partially triangulated $(\theta/4 \times \theta/4)$-grid.*

Proof. By Theorem 5.2, G has a respectful tangle of order θ. Let $A(R_G)$ be the set of vertices, edges, and regions (collectively, *atoms*) in the radial graph R_G. According to [146, Section 9] (see also [147]), the existence of a respectful tangle makes it possible to define a metric d on $A(R_G)$ as follows:

1. If $a = b$, then $d(a, b) = 0$.

2. If $a \neq b$, and a and b are interior to a contractible closed walk in the radial graph of length less than 2θ, then $d(a, b)$ is half the minimum length of such a walk. (Here by *interior* we mean the direction in which the walk can be contracted.)

3. Otherwise, $d(a, b) = \theta$.

Assume for simplicity that θ is even and that $\theta \geq 4$. Let c be any vertex in G. For $0 \leq i < \theta/2$, define Z_{2i} to be the union of all atoms of distance at most $2i$ from c (where distance is measured according to the metric d). (Notice that, in radial graphs, all closed walks have even length.) By [146, (8.10)], Z_{2i} is a nonempty simply connected set, for all i. (A subset of a surface is *simply connected* if it is connected and has no noncontractible closed curves.) Thus, the boundary $\mathbf{bd}(Z_{2i})$ of each Z_{2i} is a closed walk in the radial graph.

We claim that the closed walks $\mathbf{bd}(Z_{2i})$ and $\mathbf{bd}(Z_{2i+2})$ are vertex-disjoint. Consider any atom a on $\mathbf{bd}(Z_{2i})$ and an adjacent atom b outside Z_{2i}. The distance between a and b is 2 because there is a length-2 closed walk connecting them, doubling the edge (a, b). By Theorem 9.1 of [146], the metric d satisfies the triangle inequality, and hence $d(c, b) \leq d(c, a) + 2 = 2i + 2$. In fact, this bound must hold with equality, because $b \notin Z_{2i}$. Therefore, every atom a on $\mathbf{bd}(Z_{2i})$ is surrounded on the exterior of Z_{2i} by atoms at distance exactly $2i + 2$ from c, so $\mathbf{bd}(Z_{2i})$ is strictly enclosed by $\mathbf{bd}(Z_{2i+2})$.

Consider the "annulus" $\mathcal{A} = (Z_{2\theta-2} - Z_\theta) \cup \mathbf{bd}(Z_{2\theta-2}) \cup \mathbf{bd}(Z_\theta)$, which includes the boundary $\mathbf{bd}(\mathcal{A}) = \mathbf{bd}(Z_{2\theta-2}) \cup \mathbf{bd}(Z_\theta)$. We claim that there are at least $\theta/2$ vertex-disjoint paths in the radial graph within $\mathcal{A}$ connecting vertices in $\mathbf{bd}(Z_\theta)$ to vertices in $\mathbf{bd}(Z_{2\theta-2})$. By Menger's Theorem, the contrary implies the existence of a cut in $\mathcal{A}$ of size less than $\theta/2$ separating the two sets, which implies the existence of a cycle of length less than θ, but such a cycle must be contained in Z_θ.

Now we form a $(\theta/2 \times \theta/2)$-grid in the radial graph. The row lines in the grid are formed by taking, for each $i = \theta, \theta+2, \theta+4, \ldots, 2\theta-2$, the unique simple cycle that encloses c and that is a subset of the closed walk $\mathbf{bd}(Z_{2i})$, The column lines in the grid are formed by the $\theta/2$ vertex-disjoint paths found above. Therefore, we obtain a subdivision of the $(\theta/2 \times \theta/2)$-grid as a subgraph of the radial graph.

Finally, we transform this grid into a $(\theta/4 \times \theta/4)$-grid in the original graph G. Each grid edge in the radial graph corresponds in the original graph to a sequence of faces surrounding the edge. We replace this grid edge by the upper half of each face. In this way, each row line in the radial graph maps in the original graph to a curve above this row line. Two adjacent mapped row lines may touch but cannot properly cross, so row lines of distance 2 or more in the grid cannot overlap when mapped to the original graph. Thus, by discarding the odd-numbered row lines, and similarly for the columns, we obtain a subdivision of the $(\theta/4 \times \theta/4)$-grid in the original graph. Because each Z_{2i} was simply connected, the grid is embedded in a simply connected subset of Σ, so if we apply contractions without deletions, we obtain a partially triangulated grid. $\square$

5.2 Bidimensional Parameters and Bounded-Genus Graphs

In this section, we define a general framework of parameterized problems for which subexponential algorithms with small constants can be obtained. Our framework is sufficiently broad that an algorithmic designer needs to check only two simple properties of any desired parameter to determine the applicability and practicality of our approach.

5.2.1 Definitions

Recall from Section 5.1.2 that a *partially triangulated* $(r \times r)$*-grid* is any planar graph obtained by adding edges between pairs of nonconsecutive vertices on a common face of a planar embedding of an $(r \times r)$-grid.

Definition 5.4. *A* parameter P *is any function mapping graphs to nonnegative integers. The* parameterized problem associated with P *asks, for some fixed* k*, whether* $P(G) \leq k$ *for a given graph* G*.*

Definition 5.5. *A parameter* P *is* minor bidimensional with density δ *if*

1. *contracting or deleting an edge in a graph* G *cannot increase* $P(G)$*, and*

2. *for the* $(r \times r)$*-grid* R*,* $P(R) = (\delta r)^2 + o((\delta r)^2)$*.*

A parameter P *is called* contraction bidimensional with density δ *if*

1. *contracting an edge in a graph* G *cannot increase* $P(G)$*,*

2. *for any partially triangulated* $(r \times r)$*-grid* R*,* $P(R) \geq (\delta r)^2 + o((\delta r)^2)$*, and*

3. δ *is the smallest real number for which this inequality holds.*

In either case, P *is called* bidimensional*. The* density δ *of* P *is the minimum of the two possible densities (when both definitions are applicable). We call the sublinear function* $f(x) = o(x)$ *in the bound on* $P(R)$ *the* residual function *of* P*.*

Notice that density assigns a positive real number, typically at most 1, to any bidimensional parameter. Interestingly, this assignment defines a total order on all such parameters.

5.2.2 Examples

Many parameters are bidimensional. Here we mention just a few. Examples of minor-bidimensional parameters are

Vertex cover. A *vertex cover* of a graph G is a set C of vertices such that every edge of G has at least one endpoint in C. The *vertex-cover problem* is to find a minimum-size vertex cover in a given graph G. The corresponding parameter, the size of a minimum vertex cover, is minor bidimensional with density $\delta = 1/\sqrt{2}$. (Roughly half the vertices must be in any vertex cover of the grid, and one color class in a vertex 2-coloring of the grid is a vertex cover.)

Feedback vertex set. A *feedback vertex set* of a graph G is a set U of vertices such that every cycle of G passes through at least one vertex of U. The size of a minimum feedback vertex size is a minor-bidimensional parameter with density $\delta \in [1/2, 1/\sqrt{2}]$. ($\delta \geq 1/2$ because there are $r^2/4 + o(r^2)$ vertex-disjoint squares in the $(r \times r)$-grid, each of which must be broken; $\delta \leq 1/\sqrt{2}$ because it suffices to remove one color class in a vertex 2-coloring of the grid.)

Minimum maximal matching. A *matching* in a graph G is a set E' of edges without common endpoints. A matching in G is *maximal* if it is contained by no other matching in G. The size of a minimum maximal matching is a minor-bidimensional parameter with density $\delta \in [1/\sqrt{8}, 1/\sqrt{2}]$. ($\delta \geq 1/\sqrt{8}$ because any maximal matching must include at least one edge interior to any 3×4 subgrid, and there are $r^2/8 + o(r^2)$ interior-disjoint 3×4 subgrids; $\delta \leq 1/\sqrt{2}$ because the number of edges in a matching is at most $r^2/2$.)

Examples of contraction-bidimensional parameters are

Dominating set. A *dominating set* of a graph G is a set D of vertices of G such that each of the vertices of $V(G) - D$ is adjacent to at least one vertex of D. The size of a minimum dominating set is a contraction-bidimensional parameter with density $\delta = 1/3$. ($\delta \geq 1/3$ because every vertex dominates at most 9 vertices; $\delta \leq 1/3$ because there is a triangulation of the $(r \times r)$-grid with dominating set of size $r^2/9 + o(r^2)$.)

Edge dominating set. An *edge dominating set* of a graph G is a set D of edges of G such that every edge in $E(G) - D$ shares at least one endpoint with some edge in D. The size of a minimum edge domainting set is a contraction-bidimensional parameter with density $\delta = 1/\sqrt{14}$. ($\delta \geq 1/\sqrt{14}$ because every edge in a triangulated grid dominates at most 14 edges; $\delta \leq 1/\sqrt{14}$ because size-14 neighborhoods of a diagonal edge can be tiled to form a triangulated $(r \times r)$-grid requiring $r^2/14 + o(r^2)$ dominating edges.)

Many of our results can be applied not only to bidimensional parameters but also to parameters that are bounded by bidimensional parameters. For example, the *clique-transversal number* of a graph G is the minimum number of vertices intersecting every maximal clique of G. This parameter is not contraction-bidimensional because an edge contraction may create a new maximal clique and cause the clique-transversal number to increase. On the other hand, it is easy to see that this graph parameter always exceeds the size of a minimum dominating set. In particular, this fact can be used to obtain a parameter-treewidth bound for the clique-transversal number.

Finally, it is worth mentioning that the definition of bidimensional parameters will be further extended for the general class of H-minor-free graphs in Chapter 7.

5.2.3 Subexponential Algorithms and Planar Graphs

Almost all known techniques for obtaining subexponential parameterized algorithms on planar graphs are based on the following "bounded-treewidth approach" [2, 93, 116]:

(I1) Prove that $\mathbf{tw}(G) \leq c\sqrt{P(G)}$ for some constant c;

(I2) Compute or approximate the treewidth (or branchwidth) of G;

(I3) Decide whether $P(G) \leq k$ as follows. If the treewidth is more than $c\sqrt{k}$, then the answer to the decision problem is NO. If treewidth is at most $c\sqrt{k}$, then run a standard dynamic program for graphs of bounded treewidth in $2^{O(\mathbf{tw}(G))}n^{O(1)} = 2^{O(\sqrt{k})}n^{O(1)}$ time.

All previously known ways of obtaining the most important step (I1) use rather complicated techniques based on separators. Next we give some hints why bidimensional parameters are important for the design of subexponential algorithms by showing how step (I1) can be performed for planar graphs.

For every bidimensional parameter P and $(r \times r)$-grid R, $|V(R)| = O(P(R))$, by Theorem 4.2, we have the following proposition.

Proposition 5.6. *Let P be a bidimensional parameter. Then for any planar graph G, $\mathbf{tw}(G) = O(\sqrt{P(G)})$.*

The class of bidimensional parameterized problems contains all parameters known from the literature to have subexponential parameterized algorithms for planar graphs [3, 2, 6, 50, 123, 107]. Recently, Cai et al. [48] defined a class of parameters, Planar TMIN$_1$, and proved that, for every planar graph G and parameter P in Planar TMIN$_1$, $\mathbf{tw}(G) = O(\sqrt{P(G)})$. Every problem in Planar TMIN$_1$ can be expressed as a special type of dominating-set problem on bipartite graphs. (We refer to [48] for definitions and further properties of Planar TMIN$_1$.) Using Proposition 5.6 it is possible to prove a similar result, establishing the bound $\mathbf{tw}(G) = O(\sqrt{P(G)})$ for most parameters P in Planar TMIN$_1$.

It is tempting to wonder whether every parameter admitting a $2^{O(\sqrt{k})}n^{O(1)}$-time algorithm on planar graphs is bidimensional.

5.2.4 Parameter-Treewidth Bound for Bounded-Genus Graphs

To extend Proposition 5.6 to graphs of bounded genus, more work needs to be done.

If P is a bidimensional parameter with density δ and residual function f, then we define the *normalization factor* of P to be the minimum positive number β such that $(\frac{\delta}{\beta}r)^2 \leq (\delta r)^2 + f(\delta r)$ for any $r \geq 1$.

Lemma 5.7. *Let P be a contraction (minor) bidimensional parameter with density δ. Then $P(G) < (\frac{\delta}{\beta}r)^2$ implies that G excludes the $(r \times r)$-grid as a minor (and all partial triangulations of the $(r \times r)$-grid as contractions).*

Proof. If P is minor bidimensional and H is the $(r \times r)$-grid and $H \preceq G$, then $P(H) \leq P(G)$. Because $P(H) = (\delta r)^2 + f(\delta r)$, we have that $(\frac{\delta}{\beta}r)^2 > P(G) \geq (\delta r)^2 + f(\delta r)$, which contradicts the definition of β.

If P is contraction bidimensional and H is a partial triangulation of the $(r \times r)$-grid and $H \preceq G$, then $P(H) \leq P(G)$. Because $P(H) = (\delta r)^2 + f(\delta r)$, we have that $(\frac{\delta}{\beta} r)^2 > P(G) \geq (\delta r)^2 + f(\delta r)$, which contradicts the definition of β. $\square$

Let G be a graph and let $v \in V(G)$ be a vertex. Also suppose we have a partition $\mathcal{P}_v = (N_1, N_2)$ of the set of the neighbors of v. Define the *splitting* of G with respect to v and $\mathcal{P}_v$ to be the graph obtained from G by

1. removing v and its incident edges;

2. introducing two new vertices v^1 and v^2; and

3. connecting v^i with the vertices in N_i, for $i = 1, 2$.

If H is the result of consecutive application of several such operations to some graph G, then we say that H is a *splitting* of G. If in addition the sequence of splittings never splits a vertex that was the result of a previous splitting, then we say that H is a *fair splitting* of G. The vertices v of G involved in the splittings that make up a fair splitting are called *affected vertices.*

A parameter P is *α-splittable* if, for every graph G and for each vertex $v \in V(G)$, the result G' of splitting G with respect to v satisfies $P(G') \leq P(G) + \alpha$. Many natural graph problems are α-splittable for small constants α. Examples of 1-splittable problems are dominating set, vertex cover, edge dominating set, independent set, clique-transversal set, and feedback vertex set, among many others.

For the proof of our main result on properties of bidimensional parameters, we need two technical lemmas used in induction on the genus.

It is convenient to work with *Euler genus*. The Euler genus $\mathbf{eg}(\Sigma)$ of a nonorientable surface Σ is equal to the nonorientable genus $\tilde{g}(\Sigma)$ (or the crosscap number). The Euler genus $\mathbf{eg}(\Sigma)$ of an orientable surface Σ is $2g(\Sigma)$, where $g(\Sigma)$ is the orientable genus of Σ.

The following lemma is very useful in proofs by induction on the genus. The first part of the lemma follows from [136, Lemma 4.2.4] (corresponding to a nonseparating cycle) and the second part follows from [136, Proposition 4.2.1] (corresponding to a surface-separating cycle).

Lemma 5.8. *Let G be a connected graph 2-cell embedded in a surface Σ not homeomorphic to a sphere, and let N be a noncontractible noose on G. Then there is a fair splitting G' of G affecting the set $S = \{v_1, \ldots, v_\rho\}$ of vertices of G met by N such that one of the following holds:*

1. *G' can be 2-cell embedded in a surface with Euler genus strictly smaller than $\mathbf{eg}(\Sigma)$; or*

2. *each connected component G_i of G' can be 2-cell embedded in a surface with Euler genus strictly smaller than $\mathbf{eg}(\Sigma)$ and is a contraction of some graph G_i^* obtained from G after at most ρ splittings.*

The following lemma is a direct consequence of the definition of branchwidth.

Lemma 5.9. *Let G be a graph and let G' be the splitting of a vertex in G. Then* $\mathbf{bw}(G') \leq \mathbf{bw}(G) + 1$.

Theorem 5.10. *Suppose that P is an α-splittable contraction bidimensional parameter ($\alpha \geq 0$) with density $\delta > 0$ and normalization factor $\beta \geq 1$. Then, for any graph G 2-cell embedded in a surface Σ of Euler genus $\mathbf{eg}(\Sigma)$, $\mathbf{bw}(G) \leq 4\frac{\beta}{\delta}(\mathbf{eg}(\Sigma) + 1)\sqrt{P(G) + 1} + 8\alpha(\frac{\beta}{\delta}(\mathbf{eg}(\Sigma) + 1))^2$.*

Proof. We induct on the Euler genus of Σ.

In the base case that $\mathbf{eg}(\Sigma) = 0$, Lemma 5.7 implies that, if $P(G) < (\frac{\delta}{\beta}r)^2$, then G excludes the $(r \times r)$-grid as a minor. This implication is precisely Lemma 5.7 in the case that P is minor bidimensional. If P is contraction bidimensional, then the implication follows because, if a planar graph G can be transformed to a graph H (e.g., the $(r \times r)$-grid) via a sequence of edge contractions and/or removals, then by applying only the contractions in this sequence, we obtain a partial triangulation of H as a contraction of G. Now by Theorem 4.2, if $P(G) < (\frac{\delta}{\beta}r)^2$, then $\mathbf{bw}(G) \leq 4r - 6$. If we set $r = \lfloor \frac{\beta}{\delta}\sqrt{P(G)} \rfloor + 1$, we have that $\mathbf{bw}(G) \leq 4\lfloor \frac{\beta}{\delta}\sqrt{P(G)} \rfloor - 2$. Because $\alpha, \beta, \delta \geq 0$, the induction base follows.

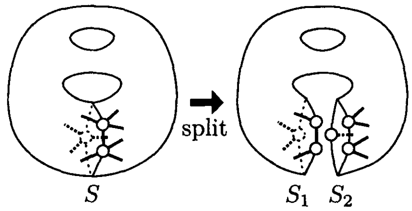

Figure 5-1: Splitting a noose.

Suppose now that $\mathbf{eg}(\Sigma) \geq 1$ and that the induction hypothesis holds for any graph 2-cell embedded in a surface with Euler genus less than $\mathbf{eg}(\Sigma)$. Let G be a graph embedded in Σ. We set $k = P(G)$ and claim that the representativity of G is at most $4\lfloor \frac{\beta}{\delta}\sqrt{k+1} \rfloor$. Lemma 5.7 implies that, if $k < (\frac{\delta}{\beta} r)^2$, then G excludes any triangulation of the $(r \times r)$-grid as a contraction. By the contrapositive of Lemma 5.3, this implies that the representativity of G is less than $4r$. If we set $r = \lfloor \frac{\delta}{\beta}\sqrt{k+1} \rfloor + 1$, we have that the representativity of G is at most $4\lfloor \frac{\beta}{\delta}\sqrt{k+1} \rfloor$. Let N be a minimum size non-contractible noose N on Σ meeting ρ vertices of G where $\rho \leq 4\lfloor \frac{\beta}{\delta}\sqrt{k+1} \rfloor$. By Lemma 5.8, there is a fair splitting along the vertices met by N such that either Condition 1 or Condition 2 holds; see Figure 5-1. Let G' be the resulting graph and let Σ' be a surface such that $\mathbf{eg}(\Sigma') \leq \mathbf{eg}(\Sigma) - 1$ and every component of G' is 2-cell embedable in Σ'. We claim that, given either Condition 1 or Condition 2, $\mathbf{bw}(G') \leq 4\frac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+\alpha\rho+1} + 8\alpha(\frac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2$.

Given Condition 1, we apply the induction hypothesis to G' and get that $\mathbf{bw}(G') \leq 4\frac{\beta}{\delta}(\mathbf{eg}(\Sigma')+1)\sqrt{P(G')+1} + 8\alpha(\frac{\beta}{\delta})^2(\mathbf{eg}(\Sigma')+1)^2$. Because G' is obtained from G after at most ρ splittings and P is an α-splittable parameter, we have $P(G') \leq k + \alpha\rho$. Because $\mathbf{eg}(\Sigma') \leq \mathbf{eg}(\Sigma) - 1$, we obtain $\mathbf{bw}(G') \leq 4\frac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+\alpha\rho+1} + 8\alpha(\frac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2$.

Given Condition 2, we apply the induction hypothesis to each of the connected components of G. Let G_i be such a component. We get that $\mathbf{bw}(G_i) \leq 4\frac{\beta}{\delta}(\mathbf{eg}(\Sigma')+1)\sqrt{P(G_i)+1} + 8\alpha(\frac{\beta}{\delta})^2(\mathbf{eg}(\Sigma')+1)^2$. Because G_i is a contraction of some graph G_i^* obtained from G after at most ρ splittings and P is an α-splittable parameter, we get that $P(G_i) \leq P(G_i^*) \leq k+\alpha\rho$. Again because $\mathbf{eg}(\Sigma') \leq \mathbf{eg}(\Sigma)-1$, we have $\mathbf{bw}(G_i) \leq 4\frac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+\alpha\rho+1}+8\alpha(\frac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2$. Because $\mathbf{bw}(G') = \max_i(\mathbf{bw}(G_i))$, we obtain

$\mathbf{bw}(G') \le 4\frac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+\alpha\rho+1} + 8\alpha(\frac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2$.

Because G' is the result of at most ρ consecutive vertex splittings on G, Lemma 5.9 yields that $\mathbf{bw}(G) \le \mathbf{bw}(G') + \rho$. Therefore,

$$\begin{aligned}
\mathbf{bw}(G) &\le 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+\alpha\rho+1} + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2 + \rho \\
&\le 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+\alpha(4\tfrac{\beta}{\delta}\sqrt{k+1})+1} + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2 + 4\tfrac{\beta}{\delta}\sqrt{k+1} \\
&= 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{(\sqrt{k+1})(\sqrt{k+1}+4\alpha\tfrac{\beta}{\delta})} + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2 + 4\tfrac{\beta}{\delta}\sqrt{k+1} \\
&\le 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{(\sqrt{k+1}+4\alpha\tfrac{\beta}{\delta})(\sqrt{k+1}+4\alpha\tfrac{\beta}{\delta})} + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2 + 4\tfrac{\beta}{\delta}\sqrt{k+1}, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{because } \alpha,\beta,\delta \ge 0 \\
&= 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)(\sqrt{k+1}+4\alpha\tfrac{\beta}{\delta}) + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2 + 4\tfrac{\beta}{\delta}\sqrt{k+1} \\
&= 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+1} + 16\alpha(\tfrac{\beta}{\delta})^2\mathbf{eg}(\Sigma) + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma))^2 + 4\tfrac{\beta}{\delta}\sqrt{k+1} \\
&= 4\tfrac{\beta}{\delta}(\mathbf{eg}(\Sigma)+1)\sqrt{k+1} + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma)^2 + 2\mathbf{eg}(\Sigma)) \\
&= 4\tfrac{\beta}{\delta}(\mathbf{eg}(\Sigma)+1)\sqrt{k+1} + 8\alpha(\tfrac{\beta}{\delta})^2(\mathbf{eg}(\Sigma)^2 + 2\mathbf{eg}(\Sigma)+1), \quad \text{because } \alpha,\beta,\delta \ge 0 \\
&= 4\tfrac{\beta}{\delta}(\mathbf{eg}(\Sigma)+1)\sqrt{k+1} + 8\alpha(\tfrac{\beta}{\delta}(\mathbf{eg}(\Sigma)+1))^2.
\end{aligned}$$

□

Theorem 5.10 is a general theorem that applies to any α-splittable bidimensional parameter. It is worth mentioning that the "α-splittablity constraint" in Theorem 5.10 can be removed by changing slightly the definition of (contraction) bidimensional parameters for bounded-genus graphs to match with the definition mentioned in Section 1.4 (see [73] for the details).

For minor-bidimensional parameters, the bound for branchwidth can be further improved.

Theorem 5.11. *Suppose that P is a minor-bidimensional parameter with density $\delta \le 1$ and normalization factor $\beta \ge 1$. Then, for any graph G 2-cell embedded in a surface Σ of Euler genus* $\mathbf{eg}(\Sigma)$, $\mathbf{bw}(G) \le 4\frac{\beta}{\delta}(\mathbf{eg}(\Sigma)+1)\sqrt{P(G)+1}$.

Proof. The proof is similar to the proof of Theorem 5.10. The only difference is that, instead of a fair splitting along the vertices of a minimum-size non-contractible noose, we just remove vertices of the noose from the graph. Because the parameter is minor bidimensional, the parameter cannot increase by this operation. The rest of the proof proceeds as before. Let G be a graph 2-cell embedded in a surface Σ of Euler genus $\mathbf{eg}(\Sigma)$, and let $k = P(G)$. We have the following substantially simpler inequality than the one in Theorem 5.10:

$$\mathbf{bw}(G) \leq 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+1} + \rho \leq 4\tfrac{\beta}{\delta}\mathbf{eg}(\Sigma)\sqrt{k+1} + 4\tfrac{\beta}{\delta}\sqrt{k+1} = 4\tfrac{\beta}{\delta}(\mathbf{eg}(\Sigma)+1)\sqrt{k+1}$$

□

5.2.5 Combinatorial Results and Further Improvements

As a consequence of Theorem 5.11, we establish an upper bound on the treewidth (or branchwidth) of a bounded-genus graph that excludes some planar graph H as a minor.

As part of their seminal Graph Minors series, Robertson and Seymour proved the following:

Theorem 5.12. [144] *If G excludes a planar graph H as a minor, then the branchwidth of G is at most b_H and the treewidth of G is at most t_H, where b_H and t_H are constants depending only on H.*

The current best estimate of these constants is the exponential upper bound $t_H \leq 20^{2(2|V(H)|+4|E(H)|)^5}$ [151]. However, it is known that planar graphs can be excluded "quickly" from planar graphs. More precisely, the following result says that, for planar graphs, the constants depend only linearly on the size of H:

Theorem 5.13. [151] *If G is planar and excludes a planar graph H as a minor, then the branchwidth of G is at most $4(2|V(H)| + 4|E(H)|) - 3$.*

Essentially the same proofs of Theorems 5.10 and 5.11 yield the following generalization of Theorem 4.2 for graphs of bounded genus. In fact, though, we can prove the following result directly from Theorem 5.11.

Theorem 5.14. *If G is a graph of Euler genus* $\mathbf{eg}(G)$ *with branchwidth more than* $4r(\mathbf{eg}(G)+1)$*, then G has the $(r \times r)$-grid as a minor.*

Proof. Consider the parameter $\xi(G) = \max\{r^2 \mid G \text{ has an } (r \times r)\text{-grid as a minor}\}$. This parameter never increases when taking minors, and has value r^2 on the $(r \times r)$-grid, so is minor bidimensional with density 1 and normalization factor 1. If G excludes the $(r \times r)$-grid as a minor, then $\xi(G) < r^2$, so $\xi(G) \leq r^2 - 1$. By Theorem 5.11, we have that $\mathbf{bw}(G) \leq 4(\mathbf{eg}(G)+1)\sqrt{\xi(G)+1} \leq 4(\mathbf{eg}(G)+1)r$, proving the contrapositive of the theorem. □

Following the proofs of Theorems 5.10 and 5.11 we are also able to quickly exclude any planar graph from bounded-genus graphs. In other words, we generalize Theorem 5.13 as follows:

Theorem 5.15. *If G is a graph of Euler genus* $\mathbf{eg}(G)$ *that excludes a planar graph H as a minor, then its branchwidth is at most* $4(2|V(H)| + 4|E(H)|)(\mathbf{eg}(G)+1)$.

5.2.6 Algorithmic Consequences

As we already discussed, the combinatorial upper bounds for branchwidth/treewidth are used for constructing subexponential parameterized algorithms as follows. Let G be a graph and P be a parameterized problem we need to solve on G. First one constructs a branch/tree decomposition of G that is optimal or "almost" optimal. A $(\theta, \gamma, \lambda)$-*approximation scheme* for branchwidth/treewidth consists of, for every w, an $O(2^{\gamma w} n^{\lambda})$-time algorithm that, given a graph G, either reports that G has branchwidth/treewidth at least w or produces a branch/tree decomposition of G with width at most θw. For example, the current best schemes are a $(3+2/3, 3.698, 3+\varepsilon)$-approximation scheme for treewidth [11] and a $(3, \lg 27, 2)$-approximation scheme for branchwidth [148].

If the branchwidth/treewidth of a graph is "large", then combinatorial upper bounds come into play and we conclude that P has no solution on G. Otherwise we run a dynamic program on graphs of bounded branchwidth/treewidth and compute $P(G)$.

Thus we conclude with the main algorithmic result of this section:

Theorem 5.16. *Let P be a bidimensional parameter with density δ and normalization factor β. Suppose P is either minor bidimensional, in which case we set $\mu = 0$, or P is contraction bidimensional and α-splittable, in which case we set $\mu = 2$. Suppose that there is an algorithm for the associated parameterized problem that runs in $O(2^{aw}n^b)$ time given a tree/branch decomposition of the graph G with width w. Suppose also that we have a $(\theta, \gamma, \lambda)$-approximation scheme for treewidth/branchwidth. Set $\tau = 1$ in the case of branchwidth and $\tau = 1.5$ in the case of treewidth. Then the parameterized problem asking whether $P(G) \leq k$ can be solved in $O(2^{\max\{a\theta,\gamma\}\tau 4\frac{\beta}{\delta}(g(G)+1)(\sqrt{k+1}+\mu\alpha\frac{\beta}{\delta}(g(G)+1))}n^{\max\{b,\lambda\}})$ time.*

The existence of an $O(2^{aw}n^b)$-time algorithm for treewidth/branchwidth w holds for many examples of bidimensional parameters with small values of a and b; see [2, 6, 50, 74, 93, 123, 94]. Observe that the correctness of our algorithms is simply based on Theorems 5.10 and 5.11, despite their nonalgorithmic natures, and $(\theta, \gamma, \lambda)$-approximation scheme for branch/tree decomposition. We note that the time bounds we provide do not contain any hidden constants, and the constants are reasonably low for a broad collection of problems covering all the problems for which $2^{O(\sqrt{k})}n^{O(1)}$-time algorithms already exist.

5.3 *H*-Minor-Free Graphs

In this section we demonstrate how the results on graphs of bounded genus can be generalized on graphs with excluded minors. More precisely we show that, given the tree decompositions guaranteed by Theorems 1.3 and 1.4, we can obtain efficient algorithms for problems on H-minor-free graphs. Although our main development is in terms of dominating set, our approach can be viewed as a guideline for solving other problems on H-minor-free graphs. Some further results in this direction are described in Section 5.4.

5.3.1 Almost-Embeddable Graphs and r-Dominating Set

In order to treat each term separately in the clique-sum decomposition of an H-minor-free graph, we need to solve a more general problem than dominating set. This r-dominating set problem, which also arises in facility location, is also contraction-bidimensional. This property enables us to obtain a parameter-treewidth bound for this problem as well.

Definition 5.17. *Let G be a graph. A subset $D \subseteq V(G)$ of vertices* r-dominates *another subset $S \subseteq V(G)$ of vertices if each vertex in S is at distance at most r from a vertex in D. We say that D is an* r-dominating set *if it r-dominates $V(G)$.*

We need the following result proved in [64].

Lemma 5.18. *(*[64], see also Chapter 4) Let $\rho, k, r \geq 1$ be integers and G be a planar graph having an r-dominating set of size k and containing a $(\rho \times \rho)$-grid as a minor. Then $k \geq (\frac{\rho-2r}{2r+1})^2$.

In other words, Lemma 5.18 says that, for any fixed r, r-dominating set is a bidimensional parameter. It is also easy to see that it is 1-splittable. Thus Theorem 5.10 yields the following lemma.

Lemma 5.19. *For any constant r, if a graph G of genus g has an r-dominating set of size at most k, then the treewidth of G is $O(g\sqrt{k} + g^2)$.*

Now we extend this result to apex-free h-almost-embeddable graphs. Before expressing this result, we need the following slight modification of [103, Lemma 2].

Lemma 5.20. *Let $G = G_0 \cup G_1 \cup \cdots \cup G_h$ be an apex-free h-almost-embeddable graph. For $1 \leq i \leq h$, let $(\mathcal{B}_u)_{u \in U_i}$ be the path decomposition of vortex G_i of width at most h. Suppose that, for each $1 \leq i \leq h$, the vertices $U_i = \{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$ form a path in G_0. Then $\mathbf{tw}(G) \leq (h^2+1)(\mathbf{tw}(G_0)+1)-1$.*

Proof. Let $\mathcal{B}$ be a bag of a tree decomposition of G_0 of minimum width $\mathbf{tw}(G_0)$. For each index $1 \leq i \leq h$, and for each vertex $u \in \mathcal{B} \cap U_i$, we add to $\mathcal{B}$ the corresponding bag $\mathcal{B}_u$ of the path decomposition of G_i. The size of each $\mathcal{B}_u$ is at most h, and the original size of $\mathcal{B}$ is also at most $\mathbf{tw}(G_0) + 1$. Thus such additions increase the size of $\mathcal{B}$ by at most $h^2(\mathbf{tw}(G_0) + 1)$. Performing these additions for all bags $\mathcal{B}$ of a tree decomposition increases the maximum bag size from $\mathbf{tw}(G_0)+1$ to $(h^2+1)(\mathbf{tw}(G_0)+1)$. It can be easily seen that the resulting set of bags $\mathcal{B}$ form a tree decomposition of G, because each U_i forms a path in G_0. □

Lemma 5.21. *Let r be a constant and let $G = G_0 \cup G_1 \cup \cdots \cup G_h$ be an apex-free h-almost-embeddable graph on a surface Σ of genus g. Let k be the size of a set $D \subseteq V(G)$ that r-dominates $V(G_0)$. Then $\mathbf{tw}(G) = O(h^2(g\sqrt{k+h} + g^2))$. In particular, for fixed g and h, $\mathbf{tw}(G) = O(\sqrt{k})$.*

Proof. For each $1 \leq i \leq h$, let $(\mathcal{B}_u)_{u \in U_i}$ be the path decomposition of vortex G_i, where $U_i = \{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$. Let G'_0 be the graph obtained from G_0 by adding new vertices $C = \{c_1, c_2, \cdots, c_h\}$ and edges (c_i, u_i^j) and (u_i^j, u_i^{j+1}) (where $j+1$ is treated modulo m_i) for all $1 \leq i \leq h$ and $1 \leq j \leq m_i$. Because G_0 is embeddable in Σ, G'_0 is also embeddable in Σ. G'_0 has an r-dominating set of size at most $k+h$, namely, $(D \cap V(G_0)) \cup C$. By Lemma 5.19, $\mathbf{tw}(G'_0) = O(g\sqrt{k+h} + g^2)$. Also, in G'_0, the vertices U_i, $1 \leq i \leq h$, form a path. By Lemma 5.20, the treewidth of $G' = G'_0 \cup G_1 \cup \cdots \cup G_h$ is $O(h^2(g\sqrt{k+h}+g^2))$. Finally, because G is a subgraph of G', $\mathbf{tw}(G) \leq \mathbf{tw}(G')$. □

5.3.2 H-Minor-Free Graphs and Dominating Set

Now that we have an understanding of r-dominating set in apex-free almost-embeddable graphs, we return to the original problem of dominating set in the more general setting of H-minor-free graphs. For this section we use the notation G^* for the entire H-minor-free graph so that the primary object of interest, an almost-embeddable piece of G^*, can be referred to as G. The main result of this section is the following algorithmic result.

Theorem 5.22. *One can test whether an H-minor-free graph G^* has a dominating set of size at most k in time $2^{O(\sqrt{k})}n^{O(1)}$, where the constants in the exponents depends* *on H.*

Before mentioning the proof of the above theorem, we need some definitions and lemmas.

Definition 5.23. *Consider a clique-sum decomposition of an H-minor-free graph G^* according to Theorem 1.3, organized into a tree structure (T, χ) as described in Section 1.3.3. Let G be one term in the clique-sum decomposition of G^* that is an h-almost embeddable on a surface of genus g, with apex set X. If we remove from T the node of T corresponding to term G, we obtain a forest T' of p subtrees; let $G_1, G_2, \ldots, G_p$ denote the clique-sums of the terms corresponding to the nodes in each connected component of T'. We say that G is* clique-summed *with each G_i, $1 \le i \le p$, with* join set *$W_i = V(G) \cap V(G_i)$. Because the clique-sums are at most h-sums, $|W_i| \le h$. A clique W_i is called* fully dominated *by a subset $S \subseteq V(G)$ of vertices in G if $V(G_i) - X \subseteq N_{G^*}(S)$; otherwise, clique W_i is called* partially dominated *by S. A vertex v of G is* fully dominated *by a set S if $N_{G^*[V(G)-X]}[v] \subseteq N_{G^*}(S)$.*

Note that the only edges that can appear in G but not in G^* are the edges among vertices of W_i, $1 \le i \le p$.

Theorem 5.24. *Let G be an h-almost embeddable on a surface of genus g in a clique-sum decomposition of a graph G^*. Suppose G is clique-summed with graphs $G_1, \ldots, G_p$ via join sets $W_1, \ldots, W_p$, where $|W_i| \le h$, $1 \le i \le p$. Suppose G^* has a dominating set of size at most k. Then there is a subset $S \subseteq V(G)$ of size at most h such that, if we form the graph $\hat{G}$ by removing all fully dominated vertices that are not included in any partially dominated clique W_i from G, then $\mathbf{tw}(\hat{G}) = O(h^2 g \sqrt{k+h} + g^2) = O(\sqrt{k})$.*

Proof. Suppose X is the set of apices in G, so that $G - X$ is an apex-free h-almost embeddable graph. Let D be a dominating set of size k of G^* and let $S = X \cap D$. We claim that S is our desired set. The rest of the proof is as follows: we construct a set $\hat{D}$ of size at most k for $\hat{G} - X$ which 2-dominates every vertex v of $\hat{G} - X$ which is not included in any vortex. Then since $\hat{G} - X$ is an apex-free h-almost-embeddable on a surface of genus g with a 2-dominating-type set of size at most k desired by

Lemma 5.21, it has treewidth at most $O(h^2 g\sqrt{k+h}+g^2)$. Then we can add vertices of X to all bags and still have a tree decomposition of width $O(h^2 g\sqrt{k+h}+g^2)$, as desired. We construct $\hat{D}$ from D as follows. First, we set $\hat{D} = D \cap V(G)$. For each $1 \leq i \leq p$, if $D \cap (V(G_i) - W_i) \neq \emptyset$ and $W_i \not\subseteq X$, we add an arbitrary vertex $w \in W_i - X$ to $\hat{D}$. Here we say a vertex v of D is *mapped* to a vertex w of $\hat{D}$ if $v = w$ or if $v \in D \cap (V(G_i) - W_i)$ and vertex $w \in W_i - X$ is the one that we have added to $\hat{D}$. One can easily observe that since each new vertex in $\hat{D}$ is in fact accounted by a unique vertex in D, $|\hat{D}| \leq k$. It only remains to show that D is a 2-dominating set for $\hat{G} - X$. If a vertex $v \in V(\hat{G}) - X$ is not fully-dominated, then there exists a vertex $w \in N_G[v]$ which is not dominated by S and thus not dominated by X (since $S = D \cap X$). It means v is 2-dominated by a vertex u of $\hat{G} - X$ which dominates w (we note that u can be originally a vertex u' in $(V(G_i) - W_i) \cap D$ which is mapped to u in $\hat{D}$). Also, we note that for each clique W_i in which there is a mapped vertex of D, this vertex dominates all vertices of $W_i - X$ in $\hat{G} - X$ and thus we keep the whole clique $W_i - X$ in G. It only remains to show that every vertex of a partially dominated clique W_i is 2-dominated by a vertex of $\hat{G} - X$. We consider two cases: if $W_i \cap S = \emptyset$, since $V(G_i) - W_i \neq \emptyset$, there must exists a (mapped) vertex of $\hat{D}$ in $W_i - X$ and we are done. Now assume $W_i \cap S \neq \emptyset$. If $W_i \subset X$ then $W_i \cap (V(\hat{G}) - X) = \emptyset$ and we are done (since there is no clique in $\hat{G} - X$ at all.) Otherwise, there exists a vertex $W_i - X$. If $(V(G_i) - W_i) \subseteq N_{G^*}(S) \neq \emptyset$, then $V(G_i) \cap D \neq \emptyset$. Thus there exists a mapped vertex $w \in W_i - X$ and we have 1-dominated vertices of $W_i - X$. As mentioned before if $D \cap (W_i - X) \neq \emptyset$, vertices $W_i - X$ are 1-dominated and we are done. The only remaining case is the case in which there exists a vertex $w \in W_i - X$ which is dominated by a vertex $x \in V(G)$ and by assumption $w \notin N_{G^*}(S)$ (we note that in this case, there is no dominating vertex in $V(G_i) - W_i$ for any i for which $w \in W_i$.) It means vertex x is not fully dominated and thus it remains in $\hat{G}$. In addition, vertex x 2-dominates all vertices of $W_i - X$, since W_i is a clique in G and thus all vertices of $W_i - X$ are 2-dominated. This completes the proof of the theorem. □

Now, we are ready to prove Theorem 5.22.

Proof of Theorem 5.22. First, we use the $n^{O(1)}$-time algorithm of Theorem 1.4 to obtain the clique-sum decomposition of graph G^*. As mentioned before, this clique-sum decomposition can be considered as a generalized tree decomposition of G^*.

More precisely, we consider the clique-sum decomposition as a rooted tree. We try to find a dominating set of size at most k in this graph using a two-level dynamic program. Suppose a graph G is an h-almost-embeddable graph on a surface of genus g in a clique-sum decomposition of a graph G^*. Assume G is clique-summed with graphs $G_0, G_1, \ldots, G_p$ via join sets $W_0, W_1, \ldots, W_p$, where $|W_i| \leq h$, $0 \leq i \leq p$. Also assume that G_0 is considered as the parent of G and $G_1, \ldots, G_p$ are considered as children of G.

Colorings. The subproblems in our first-level dynamic program are defined by a coloring of the vertices in W_i. Each vertex will be assigned one of 3 colors, labelled 0, $\uparrow 1$, and $\downarrow 1$. The meaning of the coloring of a vertex v is as follows. Color 0 represents that vertex v belongs to the chosen dominating set. Colors $\downarrow 1$ and $\uparrow 1$ represent that the vertex v is not in the chosen dominating set. Such a vertex v must have a neighbor w in the dominating set (i.e., colored 0); we say that vertex w *resolves* vertex v. Color $\downarrow 1$ for vertex v represents that the dominating vertex w is in the subtree of the clique-sum decomposition rooted at the current graph G, whereas $\uparrow 1$ represents that the dominating vertex w is elsewhere in the clique-sum decomposition. Intuitively, the vertices colored $\downarrow 1$ have already been resolved, whereas the vertices colored $\uparrow 1$ still need to be assigned to a dominating vertex.

Locally valid colorings. A coloring of the vertices of W_i is called *locally valid* with respect to sets $S_1, S_2 \subseteq V(G)$ if the following properties hold:

- for any two adjacent vertices v and w in W_i, if v is colored 0, w is colored $\downarrow 1$; and
- if $v \in S_1 \cap W_i$, then v is colored 0; and
- if $v \in S_2 \cap W_i$, then v is not colored 0.

Our colorings are similar to that of previous work (e.g., [2]), but we use them in a new dynamic-programming framework that acts over clique-sum decompositions instead of tree decompositions.

Dynamic program subproblems. Our first-level dynamic program has one subproblem for each graph G in the clique-sum decomposition and for each coloring c of the vertices in W_0. Because each join set has at most h vertices, the number of subproblems is $O(n \cdot 3^h)$. We define $D(G, c)$ to be the size of the minimum "semi"-dominating set of the vertices in subtree rooted at G subject to the following restrictions:

1. Vertices colored $\downarrow 1$ are adjacent to at least one vertex in the dominating set. (Vertices colored $\uparrow 1$ are dominated "for free".)

2. Vertices colored 0 are precisely the vertices in the dominating set.

3. Vertices in W_0 are colored according to c.

If we solve every such subproblem, then in particular, we solve the subproblems involving the root node of the clique-sum decomposition and in which every vertex is colored 0 or $\downarrow 1$. The final dominating set of size k is given by the best solution to these subproblems.

Induction step. Suppose for each coloring c of W_i, $1 \leq i \leq p$, we know $D(G_i, c)$. If the graph G is of size at most h, then we can try all colorings in $O(3^h \cdot h^2) = O(1)$ time (where the factor of h^2 is for checking validity). Thus, we focus on almost-embeddable graphs G. First, we guess a subset X of size at most h. Then for each subset S of X, we put the vertices of S in the dominating set and forbid vertices of $X - S$ from being in the dominating set. Now we remove from G all fully dominated vertices of $G - X$ that are not included in any partially dominated clique W_i. Call the resulting graph $\hat{G}$. By Theorem 5.24, $\mathbf{tw}(\hat{G}) = O(\sqrt{k})$. We can obtain such a tree decomposition of width $3 + 2/3$ times optimum, in $2^{O(\sqrt{k})} n^{3+\varepsilon}$ time by a result of Amir [11]. All vertices absent from this tree decomposition are fully dominated and thus, in any minimum dominating set that includes S, they will not appear except the following case. It is possible that up to $|X - S| = O(h)$ vertices, which are either fully dominated or belong to $V(G_i) - W_i$ where W_i is fully dominated, appear in the dominating set to dominate vertices of $X - S$. Call the set of such vertices S'. We can guess this set S' by choosing at most h vertices among the discarded vertices that have at least one neighbor in $X - S$, and then add S' to the dominating set. On the other hand, for any partially dominated clique W_i, we know that all of its vertices are present in the tree decomposition; because they form a clique, there is a bag α_i in any tree decomposition that contains all vertices of W_i. We find α_i in our tree decomposition and map W_i and G_i to this bag. We also assume W_0 is contained in all bags, because its size is at most h. Now, for each coloring c of W_0, we run the dynamic program of Alber et al. [2] on the tree decomposition, with the restriction that the colorings of the bags are locally valid with respect to $S_1 := S \cup S'$ and $S_2 := X - S$, and are consistent with the coloring c of W_0. For each bag α_i to which we mapped G_i, we add to the cost of the bag the value $D(G_i, c')$ for the current coloring c' of W_i. Using this dynamic program, we can obtain $D(G, c)$ for each coloring c of W_0.

Running time. The running time for each coloring c of W_0 and each choice of S is $2^{O(\sqrt{k})}n$ according to [2]. We have 3^h choices for c, $O(n^{h+1})$ choices for X, $O(2^h)$ choices for S, and $O(n^{h+1})$ choices for S'. Thus the running time for this inductive step is $6^h n^{2h+2} 2^{O(\sqrt{k})}$. There are $O(n)$ graphs in the clique-sum decomposition of G. Therefore, the total running time of the algorithm is $O(6^h n^{2h+3} 2^{O(\sqrt{k})}) + n^{O(1)}$ (the latter term for creating the clique-sum decomposition), which is $2^{O(\sqrt{k})} n^{O(1)}$ as desired.

□

5.4 Concluding Remarks

We have shown how to obtain subexponential fixed-parameter algorithms for the broad class of bidimensional problems on bounded-genus graphs, and for dominating set on general H-minor-free graphs for any fixed H. Our approach can also be used to obtain subexponential algorithms for other problems on H-minor-free graphs. We now demonstrate some examples of such problems.

The first example is vertex cover, where we use the following reduction. For a graph G, let G' be the graph obtained from G by adding a path of length two between any pair of adjacent vertices. The following lemma is obvious.

Lemma 5.25. *For any K_h-minor-free graph G, $h \geq 4$, and integer $k \geq 1$,*

- *G' is K_h-minor-free, and*
- *G has a vertex cover of size $\leq k$ if and only if G' has a dominating set of size $\leq k$.*

Combining Lemma 5.25 with Theorem 5.22, we conclude that parameterized vertex cover can be solved in subexponential time on graphs with an excluded minor.

Another example is the set-cover problem. Given a collection $C = \{C_1, C_2, \ldots, C_m\}$ of subsets of a finite set $S = \{s_1, s_2, \ldots, s_n\}$, a *set cover* is a subcollection $C' \subseteq C$ such that $\bigcup_{C_i \in C'} C_i = S$. The *minimum set cover* problem is to find a cover of minimum size. For an instance (C, S) of minimum set cover, its graph G_S is a bipartite graph with bipartition (C, S). Vertices s_i and C_j are adjacent in G_S if and only if $s_i \in C_j$. Theorem 5.22 can be used to prove that minimum set cover can be solved in subexponential time when G_S is H-minor free for some fixed graph H. Specifically, for a given graph G_S, we construct an auxiliary graph A_S by adding new vertices v, u, w and making v adjacent to $\{u, w, C_1, C_2, \ldots, C_m\}$. Then

- (C, S) has a set cover of size $\leq k$ if and only if A_S has a dominating set of size $\leq k + 1$, and
- if G_S is K_h-minor-free, then A_S is K_{h+1}-minor-free.

It is reasonable to believe that Theorem 5.22 generalizes to obtain a subexponential fixed-parameter algorithm for the (k, r)-*center problem* on H-minor-free graphs Recall that the (k, r)-center problem is a generalization of the dominating-set problem in which the goal is to determine whether an input graph G has at most k

vertices (called *centers*) such that every vertex of G is within distance at most r from some center. In Chapter 4, we considered this problem for planar graphs and map graphs, and presented a generalization of dynamic programming mentioned in the proof of Theorem 5.22 to solve the (k, r)-center problem for graphs of bounded treewidth/branchwidth. This dynamic program and Theorem 5.24 can be generalized to establish the desired result for H-minor-free graphs. A consequence is that we can solve the dominating-set problem in constant powers of H-minor-free graphs, which is the most general class of graphs so far for which one can obtain the exponential speedup.

It is an open and tempting question whether our technique can be generalized to solve in subexponential time on H-minor-free graphs every problem that can be solved in subexponential time on bounded-genus graphs. Recent positive progress on this question has been made [71] (see Chapter 4). Based on our results, one can obtain subexponential algorithms for any minor-bidimensional problem on H-minor-free graphs, and for any contraction-bidimensional problem on apex-minor-free graphs. Note that these results, while general, cannot be applied directly to dominating set on H-minor-free graphs. In particular, it remains open to extend the algorithmic approaches of Section 5.3 for H-minor-free graphs to all bidimensional parameters.

We also suspect that there is a strong connection between bidimensional parameters and the existence of linear-size kernels for the corresponding parameterized problems in bounded-genus graphs. Such a linear kernel has recently been obtained for dominating set [94].

Chapter 6

Diameter and Treewidth in Minor-Closed Graph Families

Eppstein [87] introduced the *diameter-treewidth property* for a class of graphs, which requires that the treewidth of a graph in the class is upper bounded by a function of its diameter. This notion has been used extensively in a slightly modified form called the *bounded-local-treewidth property*, which requires that the treewidth of any connected subgraph of a graph in the class is upper bounded by a function of its diameter. For minor-closed graph families, which is the focus of most work in this context, these properties are identical.

The reason for introducing graphs of bounded local treewidth is that they have many similar properties to both planar graphs and graphs of bounded treewidth, two classes of graphs on which many problems are substantially easier. In particular, Baker's approach for polynomial-time approximation schemes (PTASs) on planar graphs [23] applies to this setting. As a result, PTASs are known for hereditary maximization problems such as maximum independent set, maximum triangle matching, maximum H-matching, maximum tile salvage, minimum vertex cover, minimum dominating set, minimum edge-dominating set, and subgraph isomorphism for a fixed pattern [72, 87, 110]. Graphs of bounded local treewidth also admit several efficient fixed-parameter algorithms. In particular, Frick and Grohe [96] give a general framework for deciding any property expressible in first-order logic in graphs of bounded local treewidth.

The foundation of these results is the following characterization by Eppstein [87] of minor-closed families with the diameter-treewidth property. Recall from Chapter 1 that an *apex graph* is a graph in which the removal of some vertex leaves a planar graph.

Theorem 6.1. *Let $\mathcal{F}$ be a minor-closed family of graphs. Then $\mathcal{F}$ has the diameter-treewidth property if and only if $\mathcal{F}$ does not contain all apex graphs, i.e., $\mathcal{F}$ excludes some apex graph.*

In this Chapter, we reprove this theorem with a much simpler proof. Similar to Eppstein's proof, we use the following theorems from Graph Minor Theory. The $m \times m$ *grid* is the planar graph with m^2 vertices arranged on a square grid and with edges connecting horizontally and vertically adjacent vertices.

Theorem 6.2 ([80]). *For integers r and m, let G be a graph of treewidth at least $m^{4r^2(m+2)}$. Then G contains either the complete graph K_r or the $m \times m$ grid as a minor.*

It is worth mentioning that we improve the bound in the theorem above in Chapter 8.

Theorem 6.3 ([151]). *Every planar graph H can be obtained as a minor of the $r \times r$ grid H, where $r = 14|V(H)| - 24$.*

As we show in Chapter 7, the results of this chapter makes the foundation of the bidimensionality theory in apex-minor-free graphs.

Before presenting the proof of Theorem 6.1, first we present a property of apex-minor-free graphs. 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 in the grid are called *nonboundary* vertices.

Lemma 6.4. *Let G be an H-minor-free graph for an apex graph H, let $k = 14|V(H)| - 22$, and let $m > 2k$ be the largest integer such that $\mathbf{tw}(G) \geq m^{4|V(H)|^2(m+2)}$. Then*

G can be contracted into an augmented grid *R, i.e., a $(m-2k)\times(m-2k)$ grid augmented with additional edges (and no additional vertices) such that each vertex $v \in V(R)$ is adjacent to less than $(k+1)^6$ nonboundary vertices of the grid.*

Proof By Theorem 6.2, G contains an $m \times m$ grid M as a minor. Thus there exists a sequence of edge contractions and edge/vertex deletions reducing G to M. We apply to G the edge contractions from this sequence; we ignore the edge deletions; and instead of deletion of a vertex v, we only contract v into one of its neighbors. Call the new graph G', which has the $m \times m$ grid M as a subgraph and in addition $V(G') = V(M)$.

We claim that each vertex $v \in V(G')$ is adjacent to at most k^4 vertices in the central $(m-2k)\times(m-2k)$ subgrid M' of M. In other words, let N be the set of neighbors of any vertex $v \in V(G')$ that are in M'. We claim that $|N| \le k^4$. Suppose for contradiction that $|N| > k^4$.

Let n_x denote the number of distinct x coordinates of the vertices in N, and let n_y denote the number of distinct y coordinates of the vertices in N. Thus, $|N| \le n_x \cdot n_y$. Assume by symmetry that $n_y \ge n_x$, and therefore $n_y \ge \sqrt{|N|} > k^2$.

We define the subset N' of N by removing all but one (arbitrary chosen) vertex that share a common y coordinate, for each y coordinate. Thus, all y coordinates of the vertices in N' are distinct, and $|N'| = n_y > k^2$. We discard all but k^2 (arbitrarily chosen) vertices in N' to form a slightly smaller set N''. We divide these k^2 vertices into k groups each of exactly k consecutive vertices according to the order of their y coordinates. Now we construct the minor $k \times k$ grid K as shown in Figure 6-1. Because each y coordinate is unique, we can draw long horizontal segments through every point. The k columns on the left-hand and right-hand sides of M allow us to connect these horizontal segments together into k vertex-disjoint paths, each passing through exactly k vertices of N''. These paths can be connected by vertical segments within each group. This arrangement of paths has the desired $k \times k$ grid K as a minor, where the vertices of the grid correspond to the vertices in N''.

Now, if v has been used in the contraction of a vertex v' in K, we proceed as shown in Figure 6-2. First we "lift" v' from the grid—not removing it from the graph

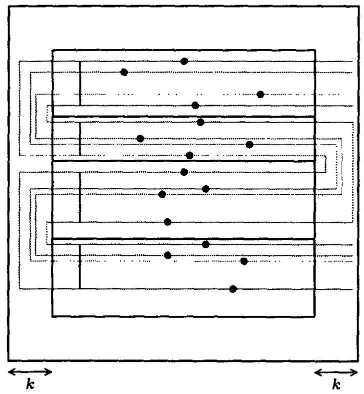

Figure 6-1: Construction of the minor $k \times k$ grid K.

per se, but marking it as "outside the grid." Then we contract the remainder of v''s column and the two adjacent columns (if they exist) into a single column. Similarly we contract the remainder of v''s row with the adjacent rows. Thus we obtain as a minor of K a $(k-2) \times (k-2)$ grid K' such that vertex v' is outside this grid and adjacent to all vertices of the grid. Now, by Theorem 6.3, because $k - 2 \geq 14|V(H)| - 24$, we can consider v' as the apex of H and obtain the planar part of H as a minor of K'. Hence the original graph G is not H-minor-free, a contradiction. This concludes the proof of the claim that $|N| \leq k^4$.

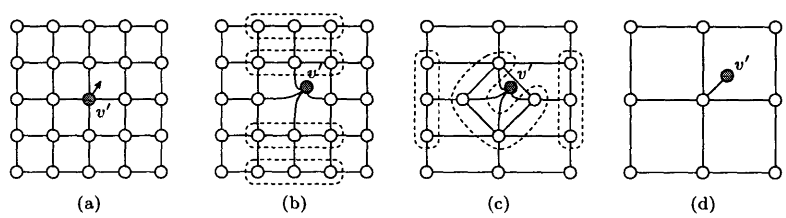

Figure 6-2: In the $k \times k$ grid K, we (a) lift the vertex v', (b) contract the adjacent columns, and (c) contract the adjacent rows, to form a $(k-2) \times (k-2)$ grid K'. Vertex v' is adjacent to all vertices in the grid, though the figure just shows four neighbors for visibility.

Finally, form a new graph R by taking graph G' and contracting all $2k$ boundary rows and $2k$ boundary columns into two boundary rows and two boundary columns (one on each side). The number of neighbors of each vertex of R that are not on the boundary is at most $(k+1)^2k^4$. The factor $(k+1)^2$ is for the boundary vertices each of which is obtained by contraction of at most $(k+1)^2$ vertices. $\square$

Now we are ready to prove Theorem 6.1.

of Theorem 6.1. One direction is easy. The apex graphs A_i, $i = 1, 2, \ldots,$ obtained from the $i \times i$ grid by connecting a new vertex v to all vertices of the grid have diameter two and treewidth $i+1$, because the treewidth of the $i \times i$ grid is i (see e.g. [79]). Thus a minor-closed family of graphs with the diameter-treewidth property cannot contain all apex graphs. Next consider the other direction. Let G be a graph from a minor-closed family $\mathcal{F}$ of graphs excluding an apex graph H. We show that the treewidth of G is bounded above by a function of $|V(H)|$ and its diameter d. Let m be the largest integer such that $\mathbf{tw}(G) \geq m^{4|V(H)|^2(m+2)}$, and let $k = 14|V(H)| - 22$. Let R be the $(m-2k) \times (m-2k)$ augmented grid obtained from G by contraction, using Lemma 6.4. Because diameter does not increase by contraction, the diameter of R is at most d. In addition, one can easily observe that the number of vertices of distance at most i from any vertex in R is at most $4r + 4r(k+1)^6 + 4r(k+1)^{12} + \cdots + 4r(k+1)^{6i} \leq 4r(k+1)^{6(i+1)}$, where $r = m - 2k$. Because the diameter is at most d, we have $4r(k+1)^{6(d+1)} \geq r^2$, i.e., $m \leq 2k + 4(k+1)^{6(d+1)}$. Thus the treewidth of G is at most $(4(k+1)^{6(d+1)} + 2k + 1)^{4|V(H)|^2(4(k+1)^{6(d+1)}+2k+3)} = O(|V(H)|^{6(d+1)})^{O(|V(H)|^{6d+8})} = 2^{O(d|V(H)|^{6d+8} \lg |V(H)|)}$, as desired. $\square$

Chapter 7

Bidimensional Parameters and Local Treewidth

As mentioned in previous chapters, the majority of results for designing FPT algorithms to solve problems such as k-vertex cover or k-dominating set in a sparse graph class $\mathcal{F}$ are based on the following lemma: every graph G in $\mathcal{F}$ where the value of the graph parameter is at most k has treewidth bounded by $t(k)$, where t is a strictly increasing function depending only on $\mathcal{F}$. With some work (sometimes very technical), a tree decomposition of width $t(k)$ is constructed and standard dynamic-programming techniques on graphs of bounded treewidth are implemented. Of course this method can not be applied for any graph class $\mathcal{F}$. For instance, the n-vertex complete graph K_n has a dominating set of size one and treewidth equal to $n-1$. So the emerging question is:

> For which (larger) graph classes and for which graph parameters can the "bounding treewidth method" be applied?

In this chapter we give a *complete* characterization of minor-closed graph families for which the aforementioned "bounding treewidth method" can be applied for a wide family of graph parameters. More precisly, we show that for a large family of *contraction-bidimensional* graph parameters, a minor-closed graph family $\mathcal{F}$ has the parameter-treewidth property if $\mathcal{F}$ has bounded local treewidth. Moreover, we show that the inverse is also correct if some simple condition is satisfied by P. In addition we show that, for a slightly smaller family of *minor-bidimensional* graph parameters, every minor-closed graph family $\mathcal{F}$ excluding some fixed graph has the parameter-treewidth property.

The proof of the main result uses the characterization of Eppstein for minor-closed families of bounded local treewidth [87] (or its simplification in Chapter 6) and Diestel et al.'s modification of the Robertson & Seymour excluded-grid-minor theorem [80]. In addition, the proof is constructive and can be used for constructing fixed-parameter algorithms to decide bidimensional graph parameters on minor-closed families of bounded local treewidth. These algorithms parallel the general fixed-parameter algorithm of Frick and Grohe [96] for properties definable in first-order logic in graph families of bounded local treewidth; our results apply e.g. to minor-bidimensional parameters definable in monadic second-order logic in nontrivial minor-closed graph families. See Section 7.4 for details.

This chapter is organized as follows. Section 7.1 contains the formal definitions of the concepts used in the chapter. Section 7.2 presents two combinatorial results which support the main result of the chapter, proved in Section 7.3. Finally, in Section 7.4 we present the algorithmic consequences of our results and we conclude with some open problems.

7.1 Definitions and Preliminary Results

We first need the following facts about treewidth. The first fact is trivial.

- For any complete graph K_n on n vertices, $\mathbf{tw}(K_n) = n - 1$.

The second fact is well known but its proof is not trivial. (See e.g., [79].)

- The treewidth of the $m \times m$ grid is m.

The next fact we need is a theorem on excluded grid minors due to Diestel et al. [80]. (See also the textbook [79].)

Theorem 7.1 ([80]). *Let r, m be integers, and let G be a graph of treewidth at least $m^{4r^2(m+2)}$. Then G contains either K_r or the $m \times m$ grid as a minor.*

Recall that a *graph parameter* P is a function mapping graphs to nonnegative integers. The *parameterized problem associated with* P asks, for a fixed k, whether $P(G) \leq k$ for a given graph G. Given a graph parameter P, we say that a graph family $\mathcal{F}$ has the *parameter-treewidth property for* P if there is a strictly increasing function t such that every graph $G \in \mathcal{F}$ has treewidth at most $t(P(G))$.

Definition 7.2. *Let $g : \mathbb{N} \to \mathbb{N}$ be a strictly increasing function. We say that a graph parameter P is* g-minor-bidimensional[1] *if*

- *Contracting an edge, deleting an edge, or deleting a vertex in a graph G cannot increase $P(G)$.*
- *For the $r \times r$ grid R, $P(R) \geq g(r)$.*

Similarly, a graph parameter P is g-contraction-bidimensional *if*

- *Contracting an edge in a graph G cannot increase $P(G)$.*
- *For any $r \times r$ augmented grid R of constant span, $P(R) \geq g(r)$.*

In the above definition, an $r \times r$ *augmented grid of span s* is an $r \times r$ *grid* with some extra edges such that each vertex is attached to at most s non-boundary vertices of the grid (see an example in Figure 7-1). Intuitively, bidimensional parameters are required to be "large" in two-dimensional grids.

We note that a g-minor-bidimensional parameter is also a g-contraction-bidimensional parameter. One can easily observe that many graph parameters such as minimum sizes of dominating set, q-dominating set (distance q-dominating set for a fixed q), vertex cover, feedback vertex set, and edge-dominating set (see exact definitions of the corresponding graph parameters in [74]) are $\Theta(r^2)$-minor- or $\Theta(r^2)$-contraction-bidimensional parameters.

[1]Closely related notions of bidimensional parameters are introduced by the authors in [63].

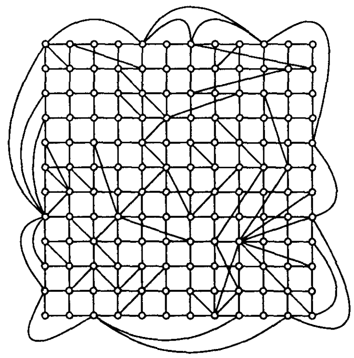

Figure 7-1: An augmented 12×12 grid with span 8.

Here, we present a theorem for minor-bidimensional parameters on general minor-closed classes of graphs excluding some fixed graphs, which plays an important role in the main result of this chapter.

Theorem 7.3. *If a g-minor-bidimensional parameter P on an H-minor-free graph G has value at most k, then* $\mathbf{tw}(G) \le 2^{4|V(H)|^2(g^{-1}(k)+2)\log(g^{-1}(k))} = 2^{O(g^{-1}(k)\log(g^{-1}(k)))}$.

Proof. Notice that $K_{|V(H)|}$ contains as a minor any graph on $|V(H)|$ vertices. Therefore we may assume that G is $K_{|V(H)|}$-minor-free. If the claimed upper bound for the treewidth of G is not correct, then Theorem 7.1 implies that G contains a $m \times m$ grid R as a minor for $m > g^{-1}(k)$. Because P is g-minor-bidimensional, $P(R) \ge g(m)$. The bidimensionality of P along with the fact that R is a minor of G yield $P(G) \ge g(m)$. Therefore, $k \ge g(m)$, a contradiction. □

Theorem 7.3 can be applied for minor-bidimensional parameters such as vertex cover or feedback vertex set.

7.2 Combinatorial Lemmas

In this section we prove two combinatorial lemmas regarding grids and graphs of bounded local treewidth. The proof of the following lemma uses the ideas in the proof of Lemma 6.1.

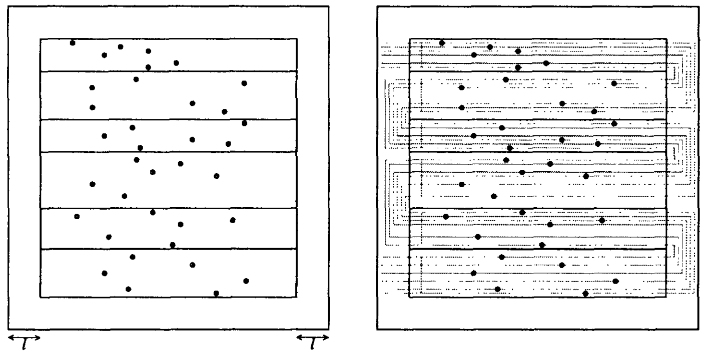

Figure 7-2: Left: The grid H, the points in S'', and their grouping. Here $\ell = 6$. Right: Construction of the minor $\ell \times \ell$ grid R passing through the points in S''.

Lemma 7.4. *Suppose we have an $m \times m$ grid H and a subset S of vertices in the central $(m - 2\ell) \times (m - 2\ell)$ subgrid H', where $s = |S|$ and $\ell = \lfloor \sqrt[4]{s} \rfloor$. Then H has as a minor the $\ell \times \ell$ grid R such that each vertex in R is a contraction of at least one vertex in S and other vertices in H.*

Proof. Let s_x denote the number of distinct x coordinates of the vertices in S, and let s_y denote the number of distinct y coordinates of the vertices in S. Thus, $s \leq s_x \cdot s_y$. Assume by symmetry that $s_y \geq s_x$, and therefore $s_y \geq \sqrt{s}$.

We define the subset S' of S by removing all but one point that share a common y coordinate, for each y coordinate. Thus, all y coordinates of the vertices in S' are distinct, and $|S'| = s_y$. We discard all but ℓ^2 vertices in S' to form a slightly smaller set S'', because $|S'| = s_y \geq \sqrt{s} \geq (\lfloor \sqrt[4]{s} \rfloor)^2 = \ell^2$. We divide these ℓ^2 vertices into ℓ groups each of exactly ℓ consecutive vertices according to the order of their y coordinates. Now we have the situation shown on the left of Figure 7-2.

We construct the minor grid R as shown on the right of Figure 7-2. Because each y coordinate is unique, we can draw long horizontal segments through every point. The ℓ columns on the left-hand and right-hand sides of H allow us to connect these horizontal segments together into ℓ vertex-disjoint paths, each passing through exactly ℓ vertices of S''. These paths can be connected by vertical segments within each group. By contracting each horizontal segment into a single vertex, and some further contraction, we can obtain the desired $\ell \times \ell$ grid R as a minor. Each vertex of this grid R is a contraction of at least one vertex in S'' (and hence in S) and other vertices in H. $\square$

Lemma 7.5. *Let $G \in \mathcal{L}(f)$ be a graph containing the $m \times m$ grid H as a subgraph, $m > 2\ell$, where $\ell = f(2) + 1$. Then the central $(m - 2\ell) \times (m - 2\ell)$ subgrid H' has the property that every vertex $v \in V(G)$ is adjacent to less than ℓ^4 vertices in H'.*

Proof. Suppose for contradiction that there is a vertex $v \in V(G)$ such that $S = N_G(v) \cap V(H)$ has size $s = |S| \geq \ell^4$. By Lemma 7.4, H' has as a minor a $\ell \times \ell$ grid R such that each vertex in R is a contraction of at least one vertex in S and other vertices in H'. If we perform these contractions and deletions in G, then v is adjacent to all vertices in R. Define $R+v$ to be the grid R plus the vertex v (if v is not already in R) and the star of connections between v and all vertices in R. Then $R + v$ is a minor of G, but has diameter 2 and treewidth $\ell \geq f(2) + 1$, a contradiction. $\square$

7.3 Main Theorem

Now we are ready to present the main result of this chapter.

Theorem 7.6. *Let P be a contraction-bidimensional parameter. A minor-closed graph class $\mathcal{F}$ has the parameter-treewidth property for P if $\mathcal{F}$ is of bounded local treewidth. In particular, for any g-contraction-bidimensional parameter P, function $f\colon \mathbb{N} \to \mathbb{N}$ and any graph $G \in \mathcal{L}(f)$ on which P has value at most k, we have $\mathbf{tw}(G) \leq 2^{O(g^{-1}(k) \log g^{-1}(k))}$. (The constant in the O notation depends on $f(1)$ and $f(2)$.)*

Proof. Let $r = f(1)+1$ and $\ell = f(2)+1$. Let $G \in \mathcal{L}(f)$ be a graph on which the graph parameter P has value k. Let m be the largest integer such that $\mathbf{tw}(G) \geq m^{4r^2(m+2)}$. Without loss of generality, we assume G is connected, and $m > 2\ell$ (otherwise, $\mathbf{tw}(G)$ is a constant because both r and ℓ are constants.) Then G has no complete graph K_r as a minor. By Theorem 7.1, G contains an $m \times m$ grid H as a minor. Thus there exists a sequence of edge contractions and edge/vertex deletions reducing G to H. We apply to G the edge contractions from this sequence, we ignore the edge deletions, and instead of deletion of a vertex v, we only contract v into one of its neighbors. Call the new graph G', which has the $m \times m$ grid H as a subgraph and in addition $V(G') = V(H)$. Because graph parameter P is contraction-bidimensional, its value on G' will not increase. By Lemma 7.5, we know that the central $(m-2\ell) \times (m-2\ell)$ subgrid H' of H has the property that every vertex $v \in V(G')$ is adjacent to less than ℓ^4 vertices in H'.

Now, suppose in graph G', we further contract all 2ℓ boundary rows and 2ℓ boundary columns into two boundary rows and two boundary columns (one on each side) and call the new graph G''. Note that here G'' and H' have the same set of vertices. The degree of each vertex of G'' to the vertices that are not on the boundary is at most $(\ell+1)^2\ell^4$, which is a constant because ℓ is a constant. Here the factor $(\ell+1)^2$ is for the boundary vertices each of which is obtained by contraction of at most $(\ell+1)^2$ vertices. Again because graph parameter P is contraction-bidimensional, its value on G'' does not increase and thus it is at most p. On the other hand, because the graph parameter is g-contraction-bidimensional, its value on graph G'' is at least $g(m-2\ell)$. Thus $g^{-1}(k) \geq m - 2\ell$, so $m = O(g^{-1}(k))$. By Theorem 7.3, the treewidth of the original graph G is at most $2^{O(g^{-1}(k) \log g^{-1}(k))}$ as desired. □

The *apex graphs* A_i, $i = 1, 2, 3, \ldots$, are obtained from the $i \times i$ grid by adding a vertex v adjacent to all vertices of the grid. It is interesting to see that, for a wide range of graph parameters, the inverse of Theorem 7.6 also holds.

Lemma 7.7. *Let P be any contraction-bidimensional parameter where $P(A_i) = O(1)$ for any $i \geq 1$. A minor-closed graph class $\mathcal{F}$ has the parameter-treewidth property for P only if $\mathcal{F}$ is of bounded local treewidth.*

Proof. The proof follows from Theorem 6.1. The apex graph A_i, has diameter ≤ 2 and treewidth $\geq i$. So a minor-closed family of graphs with the parameter-treewidth property for P cannot contain all apex graphs and hence it is of bounded local treewidth.

□

Typical examples of graph parameters satisfying Theorem 7.6 and Lemma 7.7 are dominating set and its generalization q-dominating set, for a fixed constant q (in which each vertex can dominate its q-neighborhood). These graph parameters are $\Theta(r^2)$-contraction-bidimensional and their value is 1 for any apex graph $A_i, i \geq 1$.

We can strengthen the "if and only if" result provided by Theorem 7.6 and Lemma 7.7 with the following lemma. We just need to use the fact that if the value of P is less than the value of P' then the parameter-treewidth property for P implies the parameter-treewidth property for P' as well.

Lemma 7.8. *Let P be a graph parameter whose value is lower bounded by some contraction-bidimensional parameter and let $P(A_i) = O(1)$ for any $i \geq 1$. Then a minor-closed graph class $\mathcal{F}$ has the parameter-treewidth property for P if and only if $\mathcal{F}$ is of bounded local treewidth.*

Proof. The "only if" direction is the same as in Lemma 7.7. Suppose now that P' is a contraction-bidimensional parameter where, for any graph G, $P'(G) \leq P(G)$. Applying Theorem 7.6 to P' we obtain that, if $\mathcal{F}$ is of bounded local treewidth, then $\mathcal{F}$ has the parameter-treewidth property for P', which means that there exists a strictly increasing function t such that, for any graph $G \in \mathcal{F}$, $\mathbf{tw}(G) \leq t(P'(G))$. As $P'(G) \leq P(G)$, we have that $\mathbf{tw}(G) \leq t(P(G))$ and thus $\mathcal{F}$ has the parameter-treewidth property for P. $\square$

Lemma 7.8 can be used not only for contraction-bidimensional graph parameters. As an example we mention the *clique-transversal number* of a graph, i.e., the minimum number of vertices meeting all the maximal cliques of a graph. (The clique-transversal number is not contraction-bidimensional because an edge contraction may create a new maximal clique and the value of the clique-transversal number may increase.) It is easy to see that this graph parameter always exceeds the domination number (the size of a minimum dominating set) and that any graph in A_i has a clique-transversal set of size 1.

Another application is the Π-*domination number*, i.e., the minimum cardinality of a vertex set that is a dominating set of G and satisfies some property Π in G. If this property is satisfied for any one-element subset of $V(G)$ then we call it *regular*. Examples of known variants of the parameterized dominating-set problem corresponding to the Π-domination number for some regular property Π are the following parameterized problems: the independent dominating set problem, the total dominating set problem, the perfect dominating set problem, and the perfect independent dominating set problem (see the exact definitions in [2]).

We summarize the previous observations with the following:

Corollary 7.9. *Let P be any of the following graph parameters: the minimum cardinality of a dominating set, the minimum cardinality of a q-dominating set (for any fixed q), the minimum cardinality of a clique-transversal set, or the minimum cardinality of a dominating set with some regular property* Π. *A minor-closed family of graphs $\mathcal{F}$ has the parameter-treewidth property for P if and only if $\mathcal{F}$ is of bounded local treewidth. The function $t(k)$ in the parameter-treewidth property is $2^{O(\sqrt{k}\log k)}$.*

7.4 Algorithmic Consequences and Concluding Remarks

Courcelle [61] proved a meta-theorem on graphs of bounded treewidth; he showed that, if ϕ is a property of graphs that is definable in monadic second-order logic, then ϕ can be decided in linear time on graphs of bounded treewidth. Frick and Grohe [96] partially extended this result to graphs of bounded local treewidth; they showed that, for any property ϕ that is definable in first-order logic and for any class of graphs of bounded local treewidth, there is an $O(n^{1+\varepsilon})$-time algorithm deciding whether a given graph has property ϕ, for any $\varepsilon > 0$. The constant in the O notation depends on $1/\varepsilon$, ϕ, and the local treewidth bound. However, the running time of Frick and Grohe's algorithm remains unanalyzed in terms of ϕ: their algorithm transforms ϕ into so-called "Gaifman normal form" [98] and the complexity of this transformation is unknown.

Using Theorems 7.3 and 7.6, we obtain a result along similar lines of Frick and Grohe. Specifically, consider any property that is solvable in polynomial time on graphs of bounded treewidth, e.g., properties definable in monadic second-order logic.

If the property is minor-bidimensional, we obtain a fixed-parameter algorithm on general minor-closed graph families excluding some fixed graphs; and if the property is contraction-bidimensional, we obtain a fixed-parameter algorithm on minor-closed graph families of bounded local treewidth. The differences between our result and Frick and Grohe's result are that our properties must be bidimensional but need not be definable in first-order logic, and our graph families must be minor-closed but need not have bounded local treewidth (for minor-bidimensional properties). Also, in contrast to the work of Frick and Grohe, the running time of our algorithm has an explicit bound.

Theorem 7.10. *Let P be a graph parameter such that, given a tree decomposition of width at most w for a graph G, the graph parameter can be computed in $h(w)n^{O(1)}$ time. Now, if P is a g-minor-bidimensional parameter and G belongs to a minor-closed graph family excluding some fixed graphs, or P is a g-contraction-bidimensional parameter and G belongs to a minor-closed family of graphs of bounded local treewidth, then we can compute P on G in $h(2^{O(g^{-1}(k)\log g^{-1}(k))})n^{O(1)} + 2^{2^{O(g^{-1}(k)\log g^{-1}(k))}} n^{3+\varepsilon}$ time, for any $\varepsilon > 0$.*

Proof. The algorithm is as follows. First we check whether $\mathbf{tw}(G)$ is in $2^{O(g^{-1}(k)\log g^{-1}(k))}$. By Theorems 7.3 and 7.6, if it is not, graph parameter P has value more than k on graph G. This step can be performed by Amir's algorithm [11], which for a given graph G and integer ω, either reports that the treewidth of G is at least ω, or produces a tree decomposition of width at most $(3 + \frac{2}{3})\omega$ in time $O(2^{3.698\omega} n^3 \omega^3 \log^4 n)$. Thus by using Amir's algorithm we can either compute a tree decomposition of G of size $2^{O(g^{-1}(k)\log g^{-1}(k))}$ in time $2^{2^{O(g^{-1}(k)\log g^{-1}(k))}} n^{3+\varepsilon}$, or conclude that the treewidth of G is not in $2^{O(g^{-1}(k)\log g^{-1}(k))}$.

Now if we find a tree decomposition of the aforementioned width, we can compute P on G in time $h(2^{O(g^{-1}(k)\log g^{-1}(k))})n^{O(1)}$ time. The running time of this algorithm is the one mentioned in the statement of the theorem. $\square$

For example, let G be a graph from a minor-closed family $\mathcal{F}$ of bounded local treewidth. Because the dominating set of a graph with a given tree decomposition of width at most ω can be computed in time $O(2^{2\omega}n)$ [2], Theorem 7.10 gives an algorithm which either computes a dominating set of size at most k, or concludes that there is no such a dominating set in $2^{2^{O(\sqrt{k}\log k)}}n^{O(1)}$ time. The same result holds also for computing the minimum size of a q-dominating set. Indeed, Theorem 7.10 can be applied because the q-dominating set of a graph with a given tree decomposition of width at most ω can be computed in time $O(q^{O(\omega)}n)$ [64]. Also, algorithms on graphs of bounded treewidth for clique-transversal set, and Π-domination set appeared in [50] and [2] respectively. Using these algorithms, and the fact that all these graph parameters are lower bounded by the domination number, the methodology of the proof of Theorem 7.10 can give algorithmic results for clique-transversal set and Π-domination set with the same running times as in the case of dominating set (i.e., $2^{2^{O(\sqrt{k}\log k)}}n^{O(1)}$).

Finally, it is known that the dominating set problem admits a linear size kernel on planar graphs [4]. Recently, this result was extended to graphs of bounded genus [94]. It is tempting to ask whether such a kernel exists for any minor-closed graph class of bounded local treewidth, i.e., any minor-closed graph class with the parameter-treewidth property for dominating set. The same question can be asked for other bidimensional parameters. In particular, we wonder whether there is any link between linear kernels and bidimensionality.

Chapter 8

Graphs Excluding a Fixed Minor have Grids Almost as Large as Treewidth

The $r \times r$ grid graphis the canonical planar graph of treewidth $\Theta(r)$. In particular, an important result of Robertson, Seymour, and Thomas [151] is that every planar graph of treewidth w has an $\Omega(w) \times \Omega(w)$ grid graph as a minor. Thus every planar graph of large treewidth has a grid minor certifying that its treewidth is almost as large (up to constant factors).

In their Graph Minor Theory, Robertson and Seymour [144] have generalized this result in some sense to any graph excluding a fixed minor: for every graph H and integer $r > 0$, there is an integer $w > 0$ such that every H-minor-free graph with treewidth at least w has an $r \times r$ grid graph as a minor. This result has been reproved by Robertson, Seymour, and Thomas [151], Reed [141], and Diestel, Jensen, Gorbunov, and Thomassen [80]. The best known bound on w in terms of r is as follows:

Theorem 8.1. [151, Theorem 5.8] *Every H-minor-free graph of treewidth larger than $20^{5|V(H)|^3 r}$ has an $r \times r$ grid as a minor.*

While the existence of such a relationship between treewidth and grid minors is interesting, this bound of $w = 2^{O(r)}$ is much weaker than the bound of $w = O(r)$ attainable for the special case of planar graphs. In particular, the grid they obtain from this theorem can have treewidth logarithmic in the treewidth of the original graph, which does not serve as much of a certificate of large treewidth as we have for planar graphs. The main result of this chapter is the following much tighter bound:

Theorem 8.2. *For any fixed graph H, every H-minor-free graph of treewidth w has an $\Omega(w) \times \Omega(w)$ grid as a minor.*

Thus the $r \times r$ grid is the canonical H-minor-free graph of treewidth $\Theta(r)$ for any fixed graph H. This result is best possible up to constant factors. Chapter 10 discusses the dependence of the constant factor in the Ω notation on the fixed graph H.

Our result cannot be generalized to arbitrary graphs: Robertson, Seymour, and Thomas [151] proved that some graphs have treewidth $\Omega(r^2 \lg r)$ but have grid minors only of size $O(r) \times O(r)$. The best known relation for general graphs is that having treewidth more than 20^{2r^5} implies the existence of an $r \times r$ grid minor [151]. The best possible bound is believed to be closer to $\Theta(r^2 \lg r)$ than $2^{\Theta(r^5)}$, perhaps even equal to $\Theta(r^2 \lg r)$ [151].

Our approach in the proof of Theorem 8.2 can be viewed more generally as a framework for extending combinatorial results on planar graphs to hold on H-minor-free graphs for any fixed H. The framework follows three main steps: extension from planar graphs to bounded-genus graphs, further extension to "almost-embeddable graphs", and further extension to clique sums of almost-embeddable graphs. Recall that almost-embeddable graphs are bounded-genus graphs except for a bounded number of "local areas of non-planarity", called vortices, and for a bounded number of "apex" vertices, which can have any number of incident edges that are not properly embedded. The underpinnings of this framework is the structural characterization of H-minor-free graphs in the Robertson-Seymour Graph Minor Theory [149]. Recently this framework has been used to generalize many efficient algorithms from planar graphs to H-minor-free graphs [63, 103] (see also Chapter 5). Our work shows how the framework can be applied to combinatorial results.

In addition to giving a tight bound on this basic combinatorial problem relating treewidth and grids, our result has many combinatorial consequences, each with several algorithmic consequences. For instance, one of the main consequences of our result gives the tightest possible parameter-treewidth bound for all bidimensional parameters in all possible H-minor-free graphs:

Theorem 8.3. *For any minor-bidimensional parameter P which is at least $g(r)$ in the $r \times r$ grid, every H-minor-free graph G has treewidth $\mathbf{tw}(G) = O(g^{-1}(P(G)))$. For any contraction-bidimensional parameter P which is at least $g(r)$ in an augmented $r \times r$ grid, every apex-minor-free graph G has treewidth $\mathbf{tw}(G) = O(g^{-1}(P(G)))$. In particular, if $g(r) = \Theta(r^2)$, then these bounds become $\mathbf{tw}(G) = O(\sqrt{P(G)})$.*

The proof of this theorem is identical to the proofs of Theorems 7.3 (for minor-bidimensional parameters) and 7.6 (for contraction-bidimensional parameters) except that we substitute the application of Theorem 8.1 with Theorem 8.2.

The reader is referred to Sections 1.6–1.11 to see other concequences of Theorems 8.2 and 8.3 in the bidimensionality theory.

8.1 Overview of Proof of Main Theorem

The proof of our main theorem (Theorem 8.2) is based on a series of reductions. Each reduction converts a given graph into a "simpler" graph whose treewidth is $\Omega(\mathbf{tw}(G))$.

The first reduction applies Theorem 1.3 to the original graph G, decomposing it into a clique sum of almost-embeddable graphs. By Lemma 2.4, at least one summand in this clique sum has treewidth at least $\mathbf{tw}(G)$. Therefore we can focus on this single summand of large treewidth. However, we note that this summand may not be a minor of G, and therefore it is not enough to prove that the summand has a large grid as a minor; we must deal with this issue later in the proof.

The second, trivial reduction is to remove the apices from the almost-embeddable graph. This reduction changes the treewidth by at most an additive constant. Now our almost-embeddable graph is apex-free.

The third reduction effectively removes the vortices from the apex-free almost-embeddable graph. This reduction uses that vortices have small pathwidth to conclude that the treewidth remains roughly the same. At this point the graph has bounded genus, because we have removed both apices and vortices.

Because the graph has bounded genus, it has a large grid as a minor. However, this grid is not useful: the graph is not necessarily a minor of the original graph G because, during the clique-sum decomposition, we may have introduced extra edges when the join set was completed into a clique. We call such edges *virtual edges*, and all other edges *actual edges*. One difficulty of Theorem 1.3 is that it does not guarantee that the virtual edges can be obtained by taking a minor of the original graph G, and therefore the pieces may not be minors of G. The fourth reduction overcomes this difficulty by obtaining some virtual edges by taking minors of the original graph G, and removes other virtual edges which cannot be obtained, while still preserving the treewidth up to constant factors. We call the resulting graph an *approximation graph*.

The approximation graph is both a minor of G and has bounded genus. Now we use the fact that a bounded-genus graph with treewidth w has an $\Omega(w) \times \Omega(w)$ grid as a minor. Therefore both the approximation graph and G have such a grid as a minor.

8.2 Proof of Main Theorem

In this section we prove Theorem 8.2.

First, we will need the following property about how treewidth changes during small operations to faces of a graph:

Lemma 8.4. *Consider any graph G embedded in some surface of genus g, with $\mathbf{tw}(G) = \Omega(g^2)$. If G' is the result of contracting a face of G to a point, then $\mathbf{tw}(G') \le \mathbf{tw}(G)$ and $\mathbf{tw}(G') = \Omega(\mathbf{tw}(G)/(g+1))$.*

Proof. Let f denote the face of G contracted to form G'. Because G' is a minor of G, $\mathbf{tw}(G') \le \mathbf{tw}(G)$. Consider the graph G'' formed from graph G by adding a new vertex v in the middle of face f and adding an edge connecting v to every vertex of f. This graph G'' is embedded in the same genus-g surface as G. The treewidth of G'' is at most 1 larger than the treewidth of G, by adding v to all bags of a tree decomposition of G. By [73, Theorem 2], there is a sequence of contractions that brings G'' to a graph R that is a (planar) partially triangulated $r \times r$ grid augmented with at most g additional edges, where $r = \Omega(\mathbf{tw}(G'')/(g+1)) = \Omega(\mathbf{tw}(G)/(g+1))$. Every vertex in R can be labeled by the set of vertices in G'' that were contracted to form it. Let v_R denote the vertex in R whose label includes v. For every neighbor w of v in G, the vertex w_R in R whose label includes w has distance at most 1 from v_R in R, because contractions only decrease distances. We modify the augmented partially triangulated grid R as follows. For every neighbor w of v in G for which $w_R \neq v_R$, we delete all edges incident to w_R except $\{v_R, w_R\}$, and then we contract the edge $\{v_R, w_R\}$. The resulting graph R' is a minor of R and thus of G''. If we re-order the sequence of contractions and deletions that bring G'' to R' to start with the contractions of the edges between v and the vertices of face f (which is equivalent to contracting the face f in the original graph G), then the succeeding sequence of contractions and deletions brings G' to R'. Therefore R' is a minor of G'. By ignoring every row or column of the grid in which an edge was deleted, we obtain an $(r-g-4) \times (r-g-4)$ grid minor of G'. (There may be g such rows (resp., columns) from the g additional edges, 2 from the neighborhood of v, and 2 from the boundary of the grid.) Therefore $\mathbf{tw}(G') \geq r - g - 5 = \Omega(\mathbf{tw}(G)/(g+1))$. □

Now we apply Theorem 1.3 to the original graph G, decomposing it into a clique sum of almost-embeddable graphs.

Lemma 8.5. *At least one summand in the clique sum has treewidth at least* $\mathbf{tw}(G)$.

Proof. Immediate by Lemma 2.4. □

Let G' denote a summand in the clique sum with $\mathbf{tw}(G') \geq \mathbf{tw}(G)$. For every vertex v in G', there is a corresponding vertex $f(v)$ in G by following the definition of clique sum. Each edge $\{u, v\}$ in G' may or may not have a corresponding edge

$\{f(u), f(v)\}$ in G. If the edge $\{f(u), f(v)\}$ exists in G, we say that $\{u, v\}$ is an *actual edge* in G'; otherwise, it is a *virtual edge* in G'. Virtual edges arise from removing edges from the join set during a clique sum.

Because G' is h-almost-embeddable in some bounded-genus surface, it consists of a bounded-genus graph augmented by at most h vortices and at most h apices. We remove all apices from G' to produce an apex-free h-almost-embeddable graph G''. Because adding a vertex and any collection of incident edges to a graph can increase the treewidth by at most 1, we have the following relation between the treewidths of G' and G'':

Lemma 8.6. *$\mathbf{tw}(G'') \geq \mathbf{tw}(G') - h$.*

Next we remove all vortices from G''. Let G''_0 denote the bounded-genus part of the apex-free h-almost-embeddable graph G'', and let U_i denote the set of vertices at which vortex i is attached to G''_0 (as in definition of h-almost embeddable graphs). Define $G''' = G''_0 - U_1 - U_2 - \cdots - U_h$, i.e., G''' is the result of removing all vertices from vortices in G''.

Lemma 8.7. *$\mathbf{tw}(G''') = O(\mathbf{tw}(G''))$ for $h = O(1)$.*

Proof. Suppose G'' decomposes into $G''_0 \cup G''_1 \cup G''_2 \cup \cdots \cup G''_h$ where each G''_i, $i \geq 1$, is a vortex as in definition of h-almost embeddable graphs. Define an intermediate graph $\hat{G}$ as follows. Let $U_i = \{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$ be the cyclically ordered vertices of G''_0 at which vortex G''_i is attached. We obtain $\hat{G}$ by starting from G''_0 and adding edges $\{u_i^j, u_i^{j+1}\}$ where they do not already exist, and where $j+1$ is treated modulo m_i, for each $1 \leq i \leq h$ and each $1 \leq j \leq m_i$. Because we only added a planar graph within the face corresponding to U_i, $\hat{G}$ is embeddable in the same bounded-genus surface as G''_0.

We claim that $\mathbf{tw}(G'') \leq (h+1)^2\,(\mathbf{tw}(\hat{G}) + 1)$. Consider some minimum-width tree decomposition of $\hat{G}$, and consider each bag $\mathcal{B}$ of that tree decomposition. For each u_i^j that occurs in bag $\mathcal{B}$, we add to $\mathcal{B}$ the corresponding bag $\mathcal{B}_{u_i^j}$ from the path decomposition of vortex G''_i. The resulting bags form a tree decomposition of G'' because $\{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$ are connected in a path in $\hat{G}$. By charging the $\leq h+1$

added vertices to the occurrence of u_i^j that triggered the addition, each bag increases in size by a factor at most $h+1$ for each of the h vortices. Thus the width of this tree decomposition of G'' is at most $(h(h+1))\,(\mathbf{tw}(\hat{G})+1)-1$, which is stronger than the desired claim.

Let $\hat{\hat{G}}$ be the graph resulting from $\hat{G}$ by contracting the face $\{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$ in $\hat{G}$ into a single vertex, for each i. Applying Lemma 8.4, h times, we obtain $\mathbf{tw}(\hat{\hat{G}}) = \Omega(\mathbf{tw}(\hat{G}))$ because h and the genus of the surface in which $\hat{G}$ is embedded are $O(1)$. Therefore $\mathbf{tw}(G'') = O(\mathbf{tw}(\hat{\hat{G}}))$.

Finally we delete each contracted vertex in $\hat{\hat{G}}$, which results in G'''. Thus $\mathbf{tw}(G''') \geq \mathbf{tw}(\hat{\hat{G}}) - h$, so $\mathbf{tw}(G'') = O(\mathbf{tw}(G'''))$ as desired. □

At this point the graph has bounded genus, because we have removed both apices and vortices. In the next step we deal with virtual edges. Intuitively, for each summand G' in the clique-sum decomposition of the original graph G, we construct a graph $\tilde{G}$ which is a minor of G and "approximately" preserves the virtual edges within G'. For this step we need an additional property of the clique-sum decomposition obtained in the proof of Theorem 1.3: each clique sum involves at most three vertices from each summand other than apices and vertices in vortices of that summand. This stronger form of Theorem 1.3 follows from exactly the same proof from [149]. At a high level, the proof of Theorem 1.3 ([149, Theorem 1.3]) consists of two components, [149, Theorem 2.3] and [149, Theorem 3.1]. The first part uses clique sums that involve only apices and vertices in vortices, while the second part uses clique sums that involve only three vertices in each summand (from the 3-separations). When these clique sums are combined, we may obtain clique sums involving apices, vertices in vortices, and up to three additional vertices in each summand. This fact has been confirmed independently by Seymour [154].

Definition 8.8. *Let G' be an h-almost-embeddable graph in a clique-sum decomposition of a graph G arising from Theorem 1.3. The* approximation graph *of G', denoted by $\tilde{G}$, is obtained by starting from G''', removing the virtual edges, and replacing some of them as follows. In the clique-sum decomposition of G, for each clique* *sum involving G' with the property that the join set W has $|W \cap V(G''')| > 1$, we do the following:*

1. *If $|W \cap V(G''')| = 2$, we add an edge between these two vertices.*

2. *If $|W \cap V(G''')| = 3$ and there is more than one clique sum that contains $W \cap V(G''')$ in its join set, we add all edges between pairs of vertices in $W \cap V(G''')$.*

3. *If $|W \cap V(G''')| = 3$ and there is only one clique sum that contains $W \cap V(G''')$ in its join set, we add a new vertex v inside the triangle of $W \cap V(G''')$ on the surface and then add an edge connecting v to each vertex of $W \cap V(G''')$.*

Lemma 8.9. *Let G' be an h-almost-embeddable graph in a clique-sum decomposition of a graph G arising from Theorem 1.3. The approximation graph $\tilde{G}$ of G' is a minor of G and can be embedded in the same surface as the bounded-genus part of G'.*

Proof. First, G''' with all virtual edges removed is a minor of G, because the former graph can be constructed from G by deleting all vertices not in the summand G' and deleting all apices and vertices in vortices in G'. All that remains to show is that the edges added in Cases 1–3 of Definition 8.8 can also be formed as a minor of G. We use the (trivial) additional property of the clique-sum decomposition arising from Theorem 1.3 that each summand in the clique sum is connected even after removal of the join set. (If a summand were not connected after the removal of the join set, we could rewrite the initial clique-sum decomposition by splitting the summand into a clique sum of these pieces.) Now, for each clique sum between G' and F with the property that the join set W has $|W \cap V(G''')| > 1$, we contract F down to a single vertex v adjacent to all vertices in the join set. In Case 3, this vertex v is precisely the desired vertex v inside the triangle $W \cap V(G''')$. This triangle is guaranteed to be empty in the bounded-genus part of G' in the clique-sum decomposition arising from Theorem 1.3; if this were not the case, again we could rewrite the clique-sum decomposition by splitting G' into a clique sum of two pieces. Thus the resulting graph can be embedded in the same surface as the bounded-genus part of G'. In the other two cases, we contract v into a vertex of $W \cap V(G''')$—in Case 2, we contract two different v's into two different vertices of $W \cap V(G''')$—and obtain the additional edges added to $\tilde{G}$. Finally, we delete the apices and vertices in vortices in G', and delete any other summands that had $|W \cap V(G''')| \le 1$. In the end we have contracted and deleted edges in G to obtain precisely $\tilde{G}$. □

Lemma 8.10. $\mathbf{tw}(\tilde{G}) \ge \frac{1}{3}(\mathbf{tw}(G''') + 1) - 1$.

Proof. To prove that $\mathbf{tw}(G''') \le 3(\mathbf{tw}(\tilde{G}) + 1) - 1$, we start from a minimum-width tree decomposition of $\tilde{G}$ and convert it into a tree decomposition of G'''. We need only consider Case 3 in Definition 8.8 because otherwise $\tilde{G}$ is identical to G'''. For each occurrence of an added vertex v from Case 3 in a bag $\mathcal{B}$ in the tree decomposition of $\tilde{G}$, we replace v in $\mathcal{B}$ with all three vertices from $W \cap V(G''')$. The result is a tree decomposition of G''' where each bag has increased in size by at most a factor of 3. □

By Lemma 8.9, the approximation graph $\tilde{G}$ is both a minor of G and has bounded genus. By [63, Theorem 3.5] (see also Theorem 5.14), every bounded-genus graph with treewidth $\Omega(r)$ has an $r \times r$ grid as a minor. By Lemmas 8.5, 8.6, 8.7, and 8.10, $\mathbf{tw}(\tilde{G}) = \Omega(\mathbf{tw}(G))$. Therefore $\tilde{G}$ and thus G have an $\Omega(\mathbf{tw}(G)) \times \Omega(\mathbf{tw}(G))$ grid as a minor. This concludes the proof of Theorem 8.2.

Chapter 9

Improved Approximation Algorithms for Minimum-Weight Vertex Separators and Treewidth

Given a graph $G = (V, E)$, one is often interested in finding a small "separator" whose removal from the graph leaves two sets of vertices of roughly equal size (say, of size at most $2|V|/3$), with no edges connecting these two sets. The separator itself may be a set of edges, in which case it is called a *balanced edge separator*, or a set of vertices, in which case it is called a *balanced vertex separator*. In the present work, we focus on vertex separators.

Balanced separators of small size are important in several contexts. They are often the bottlenecks in communication networks (when the graph represents such a network), and can be used in order to provide lower bounds on communication tasks (see e.g. [130, 128, 28]). Perhaps more importantly, finding balanced separators of small size is a major primitive for many graph algorithms, and in particular, for those that are based on the divide and conquer paradigm [132, 28, 129].

Certain families of graphs always have small vertex separators. For example, trees always have a vertex separator containing a single vertex. The well known planar separator theorem of Lipton and Tarjan [132] shows that every n-vertex planar graph has a balanced vertex separator of size $O(\sqrt{n})$, and moreover, that such a separator can be found in polynomial time. This was later extended to show that more general families of graphs (any family of graphs that excludes some minor, and certain geometric graphs) have small separators [101, 10, 135]. However, there are families of graphs (for example, expander graphs and the complete graph) in which the smallest separator is of size $\Omega(n)$.

Finding the smallest separator is an NP-hard problem (see, e.g. [45]). In this chapter, we study approximation algorithms that find vertex separators whose size is not much larger than the optimal separator of the input graph. These algorithms can be useful in detecting small separators in graphs that happen to have small separators, as well as in demonstrating that an input graph does not have any small vertex separator (and hence, for example, does not have serious bottlenecks for routing).

Much of the previous work on approximating vertex separators piggy-backed on work for approximating edge separators. For graphs of bounded degree, the sizes of the minimum edge and vertex separators are the same up to a constant multiplicative factor, leading to a corresponding similarity in terms of approximation ratios. However, for general graphs (with no bound on the degree), the situation is different. For example, every edge separator for the star graph has $\Omega(n)$ edges, whereas the minimum vertex separator has just one vertex. There are simple reductions from the problem of approximating edge separators to the the problem of approximating vertex separators. (For example, replace every vertex v by clique C_v on n^3 vertices, and every original edge $e = (u, v)$ by a vertex v_e connected to all vertices of C_u and C_v.) As to the reverse direction, it is only known how to reduce the problem of approximating vertex separators to the problem of approximating edge separators in *directed graphs* (a notion that will not be discussed in this chapter).

The previous best approximation ratio for vertex separators is based on the work of Leighton and Rao [129]. They presented an algorithm based on linear programming that approximates the minimum edge separator within a ratio of $O(\log n)$. They observed that their algorithm can be extended to work on directed graphs, and hence gives an approximation ratio of $O(\log n)$ also for vertex separators, using the algorithm for (directed) edge separators as a black box. More recently, Arora, Rao and

Vazirani [19] presented an improved algorithm based on semidefinite programming that approximates the minimum edge separator within a ratio of $O(\sqrt{\log n})$. Their remarkable techniques, which are a principal component in our algorithm for vertex separators, are discussed more in the following section.

In the present work, we formulate new linear and semidefinite program relaxations for the vertex separator problem, and then develop rounding algorithms for these programs. The rounding algorithms are based on techniques that were developed in the context of edge separators, but we exploit new properties of these techniques and adapt and enhance them to the case of vertex separators. Using this approach, we improve the best approximation ratio for vertex separators to $O(\sqrt{\log n})$. In fact, one can obtain an $O(\sqrt{\log \mathsf{opt}})$ approximation, where opt is the size of an optimal separator (see [90]). (An $O(\log \mathsf{opt})$ approximation can be derived from the techniques of [129].) In addition, we derive and extend some previously known results in a unified way, such as a constant factor approximation for vertex separators in planar graphs (a result originally proved in [12]), which we extend to any family of graphs excluding a fixed minor.

Before delving into more details, let us mention two aspects in which edge and vertex separators differ. One is the notion of a *minimum ratio cut*, which is an important notion used in our analysis. For edge cuts, all "natural" definitions of such a notion are essentially equivalent. For vertex separators, this is not the case. One consequence of this is that our algorithms provide a good approximation for *vertex expansion* in bounded degree graphs, but not in general graphs. This issue will be discussed in Section 9.3. Another aspect where there is a distinction between edge and vertex separators is that of the role of embeddings into L_1 (a term that will be discussed later). For edge separators, if the linear or semidefinite program relaxations happen to provide such an embedding (i.e. if the solution is an L_1 metric), then they in fact yield an optimal edge separator. For vertex separators, embeddings into L_1 seem to be insufficient, and we give a number of examples that demonstrate this deficiency. Our rounding techniques for the vertex separator case are based on embeddings with small average distortion into a line, a concept that was first systematically studied by Rabinovich [139].

As mentioned above, finding small vertex separators is a basic primitive that is used in many graph algorithms. Consequently, our improved approximation algorithm for minimum vertex separators can be plugged into many of these algorithms, improving either the quality of the solution that they produce, or their running time. Rather than attempting to provide in this chapter a survey of all potential applications, we shall present one major application, that of improving the approximation ratio for treewidth, and discuss some of its consequences.

9.1 Related Work

An important concept that we shall use is the *ratio* of a vertex separator (A, B, S). Given a weight function $\pi : V \to \mathbb{R}_+$ on vertices, and a set $S \subseteq V$ which separates G into two disconnected pieces A and B, we can define the *sparsity* of the separator by

$$\frac{\pi(S)}{\min\{\pi(A), \pi(B)\} + \pi(S)}.$$

Indeed, most of our effort will focus on finding separators (A, B, S) for which the sparsity is close to minimal among all vertex separators in G.

In the case of edge separators, there are intimate connections between approximation algorithms for minimum ratio cuts and the theory of metric embeddings. In particular, Aumann and Rabani [21] and Linial, London, and Rabinovich [131] show that one can use L_1 embeddings to round the solution to a linear relaxation of the problem. For the case of vertex cuts, we will show that L_1 embeddings (and even Euclidean embeddings) are insufficient, but that the additional structure provided by many embedding theorems does suffice. This structure corresponds to the fact that many embeddings are of *Fréchet-type*, i.e. their basic component takes a metric space X and a subset $A \subseteq X$ and maps every point $x \in X$ to its distance to A. This includes, for instance, the classical theorem of Bourgain [43].

The seminal work of Leighton and Rao [129] showed that, in both the edge and vertex case, one can achieve an $O(\log n)$ approximation algorithm for minimum-ratio cuts, based on a linear relaxation of the problem. Their solution also yields the first approximate max-flow/min-cut theorems in a model with uniform demands. The papers [131, 21] extend their techniques for the *edge case* to non-uniform demands. Their main tool is Bourgain's theorem [43], which states that every n-point metric space embeds into L_1 with $O(\log n)$ distortion.

Recently, Arora, Rao, and Vazirani [19] exhibit an $O(\sqrt{\log n})$ approximation for finding minimum-ratio edge cuts, based on a known semi-definite relaxation of the problem, and a fundamentally new technique for exploiting the global structure of the solution. This approach, combined with the embedding technique of Krauthgamer, Lee, Mendel, and Naor [124], has been extended further to obtain approximation algorithms for minimum-ratio edge cuts with non-uniform demands. In particular, by combining [19], [124], and the quantitative improvements of Lee [126], Chawla, Gupta, and Räcke [52] exhibit an $O(\log n)^{3/4}$ approximation. More recently, Arora, Lee, and Naor [18] have improved this bound almost to that of the uniform case, yielding an approximation ratio of $O(\sqrt{\log n}\log\log n)$.

On the other hand, progress on the vertex case has been significantly slower. In the sections that follow, we attempt to close this gap by providing new techniques for finding approximately optimal vertex separators.

9.2 Results and Techniques

In Section 9.3, we introduce a new semi-definite relaxation for the problem of finding minimum-ratio vertex cuts in a general graph. In preparation for applying the techniques of [19], the relaxation includes so-called "triangle inequality" constraints on the variables. The program contains strictly more than one variable per vertex of the graph, but the SDP is constructed carefully to lead to a single metric of negative type[1] on the vertices which contains all the information necessary to perform the

[1] A metric space (X, d) is said to be of *negative type* if $d(x,y) = ||f(x) - f(y)||^2$ for some map $f : X \to L_2$.

rounding.

In Section 9.4, we exhibit a general technique for rounding the solution to optimization problems involving "fractional" vertex cuts. These are based on the ability to find line embeddings with small *average* distortion, as defined by Rabinovich [139] (though we extend his definition to allow for arbitrary weights in the average). In [139], it is proved that one can obtain constant factor average distortion embeddings into the line for metrics supported on planar graphs. This is observed only as an interesting structural fact, without additional algorithmic consequences over the known average distortion embeddings into all of L_1 [140, 120]. For the vertex case, we will see that this additional structure is crucial.

Using the seminal results of [19], which can be viewed as a line embedding, we then show that the solution of the semi-definite relaxation can be rounded to a vertex separator whose ratio is within $O(\sqrt{\log n})$ of the optimal separator. In the standard SDP for minimum-ratio edge cuts (employed in the algorithms of [19]), no lower bound is known on the integrality gap. Very recent work of Khot and Vishnoi [119] shows that in the non-uniform demand case, the gap must tend to infinity with the size of the instance. In contrast, we show that our analysis is tight by exhibiting an $\Omega(\sqrt{\log n})$ integrality gap for the SDP in Section 9.6. Interestingly, this gap is achieved by an L_1 metric. This shows that L_1 metrics are not as intimately connected to vertex cuts as they are to edge cuts, and that the use of the structural theorems discussed in the previous paragraph is crucial to obtaining an improved bound.

We exhibit an $O(\log k)$-approximate max-flow/min-vertex-cut theorems for general instances with k commodities. The best previous bound of $O(\log^3 k)$ is due to [88] (they actually show this bound for directed instances with symmetric demands, but this implies the vertex case). This is proved in Section 9.5. A well-known reduction shows that this theorem implies the edge version of [131, 21] as a special case. Again, our rounding makes use of the general tools developed in Section 9.4 based on average-distortion line embeddings. In Section 9.5.2 we show that any approach based on L_1 embeddings and Euclidean embeddings, respectively, must fail since the integrality gap can be very large even for such metric spaces. Using the improved line embeddings for metrics on graphs which exclude a fixed minor [139] (based on [120] and [140]), we also achieve a constant-factor approximation for finding minimum ratio vertex cuts in these families. This answers an open problem asked in [71].

By improving the approximation ratios for balanced vertex separators in general graphs and graphs excluding a fixed minor, we improve the approximation factors for a number of problems relating to graph-theoretic decompositions such as treewidth, branchwidth, and pathwidth. For instance, we show that in any graph of treewidth k, we can find a tree decomposition of width at most $O(k\sqrt{\log k})$. If the input graph excludes some fixed minor, we give an algorithm that finds a decomposition of width $O(k)$. A discussion of these problems, along with the salient definitions, appears in Section 9.7. See Theorem 9.16 and Corollary 9.17 for a list of the problems to which our techniques apply.

Improving the approximation factor for treewidth in general graphs and graphs excluding a fixed minor to $O(\sqrt{\log n})$ and $O(1)$, respectively, answers an open problem of [71], and leads to an improvement in the running time of approximation schemes and sub-exponential fixed-parameter algorithms for several NP-hard problems on graphs excluding a fixed minor. For instance, we obtain the first polynomial-time approximation schemes (PTAS) for problems like *minimum feedback vertex set* and *connected dominating set* in such graphs (see Theorem 9.18 for a full list). Finally, our techniques yield an $O(g)$-approximation algorithm for the vertex separator problem in graphs of genus at most g. It is known that such graphs have balanced separators of size $O(\sqrt{gn})$ [101], and that these separators can be found efficiently [117] (earlier, [10] gave a more general algorithm which, in particular, finds separators of size $O(\sqrt{g^{3/2}n})$). Our approximation algorithms thus finds separators of size $O(\sqrt{g^3n})$, but when the graph at hand has a smaller separator, our algorithms perform much better than the worst-case bounds of [101, 10, 117].

9.3 A Vector Program for Minimum-Ratio Vertex Cuts

Let $G = (V, E)$ be a graph with non-negative vertex weights $\pi : V \to [0, \infty)$. For a subset $U \subseteq V$, we write $\pi(U) = \sum_{u \in U} \pi(u)$. A vertex separator partitions the graph into three parts, S (the set of vertices in the separator), A and B (the two parts that are separated). We use the convention that $\pi(A) \geq \pi(B)$. We are interested in finding separators that minimize the ratio of the "cost" of the separator to its "benefit." Here, the cost of a separator is simply $\pi(S)$. As to the benefit of a separator, it turns out that there is more than one natural way in which one can define it. The distinction between the various definitions is relatively unimportant whenever $\pi(S) \leq \pi(B)$, but it becomes significant when $\pi(S) > \pi(B)$. We elaborate on this below.

In analogy to the case of edge separators, one may wish to take the benefit to be $\pi(B)$. Then we would like to find a separator that minimizes the ratio $\pi(S)/\pi(B)$. However, there is evidence that no polynomial time algorithm can achieve an approximation ratio of $O(|V|^{\delta})$ for this problem (for some $\delta > 0$). See [90] for details.

For the use of separators in divide and conquer algorithms, the benefit is in the reduction in size of the parts that remain. This reduction is $\pi(B) + \pi(S)$ rather than just $\pi(B)$, and the quality of a separator is defined to be

$$\frac{\pi(S)}{\pi(B) + \pi(S)}.$$

This definition is used in the introduction of this Chapter, and in some other earlier work (e.g. [12]).

As a matter of convenience, we use a slightly different definition. We shall define the *sparsity* of a separator to be

$$\alpha_{\pi}(A, B, S) = \frac{\pi(S)}{\pi(A \cup S) \cdot \pi(B \cup S)}.$$

Under our convention that $\pi(A) \geq \pi(B)$, we have that $\pi(V)/2 \leq \pi(A \cup S) \leq \pi(V)$, and the two definitions differ by a factor of $\Theta(\pi(V))$.

We define $\alpha_\pi(G)$ to be the minimum over all vertex separators (A, B, S) of $\alpha_\pi(A, B, S)$. The problem of computing $\alpha_\pi(G)$ is NP-hard (see [45]). Our goal is to give algorithms for finding separators (A, B, S) for which $\alpha_\pi(A, B, S) \le O(\sqrt{\log k})\, \alpha_\pi(G)$, where $k = |\mathrm{supp}(\pi)|$ is the number of vertices with non-zero weight in G.

Before we move onto the main algorithm, let us define

$$\tilde{\alpha}_\pi(A, B, S) = \pi(S)/[\pi(A) \cdot \pi(B \cup S)].$$

Note that $\alpha_\pi(A, B, S)$ and $\alpha_\pi(A, B, S)$ are equivalent up to a factor of 2 whenever $\pi(A) \ge \pi(S)$. Hence in this case it will suffice to find a separator (A, B, S) with $\alpha_\pi(A, B, S) \le O(\sqrt{\log k})\, \tilde{\alpha}_\pi(G)$. Allowing ourselves to compare $\alpha_\pi(A, B, S)$ to $\tilde{\alpha}_\pi(G)$ rather than $\alpha_\pi(G)$ eases the formulation of the semi-definite relaxations that follow. When $\pi(S) > \pi(A)$, $\tilde{\alpha}$ no longer provides a good approximation to α. (Moreover, it is hard to approximate $\tilde{\alpha}(G)$ in this case, as shown in [90].) However, in this case the trivial separator $(\emptyset, \emptyset, V)$ has sparsity at most a constant factor larger than $\alpha(G)$, making the approximation of $\alpha(G)$ trivial.

9.3.1 The Quadratic Program

We present a quadratic program for the problem of finding min-ratio vertex cuts. All constraints in this program involve only terms that are quadratic (products of two variables). Our goal is for the value of the quadratic program to be equal to $\tilde{\alpha}_\pi(G)$. Let $G = (V, E)$ be a vertex-weighted graph, and let (A^*, B^*, S^*) be an optimal separator according to $\tilde{\alpha}_\pi(\cdot)$, i.e. such that $\tilde{\alpha}_\pi(G) = \tilde{\alpha}_\pi(A^*, B^*, S^*)$.

With every vertex $i \in V$, we associate three indicator 0/1 variables, x_i, y_i and s_i. It is our intention that for every vertex exactly one indicator variable will have the value 1, and that the other two will have value 0. Specifically, $x_i = 1$ if $i \in A^*$, $y_i = 1$ if $i \in B^*$, and $s_i = 1$ if $i \in S^*$. To enforce this, we formulate the following two sets of constraints.

Exclusion constraints. These force at least two of the indicator variables to be 0.

$$x_i \cdot y_i = 0,\ x_i \cdot s_i = 0,\ y_i \cdot s_i = 0, \text{ for every } i \in V.$$

Choice constraints. These force the non-zero indicator variable to have value 1.

$$x_i^2 + y_i^2 + s_i^2 = 1, \text{ for all } i \in V.$$

The combination of exclusion and choice constraints imply the following *integrality constraints*, which we formulate here for completeness, even though they are not explicitly included as part of the quadratic program: $x_i^2 \in \{0,1\}$, $y_i^2 \in \{0,1\}$, $s_i^2 \in \{0,1\}$, for all $i \in V$.

Edge constraints. This set of $2|E|$ constraints express the fact that there are no edges connecting A and B.

$$x_i \cdot y_j = 0 \text{ and } x_j \cdot y_i = 0, \text{ for all } (i,j) \in E.$$

Now we wish to express the fact that we are minimizing $\tilde{\alpha}_\pi(A,B,S)$ over all vertex separators (A,B,S). To simplify our presentation, it will be convenient to assume that we know the value $K = \pi(A^*) \cdot \pi(B^* \cup S^*)$. This assumption can be made without loss of generality because it suffices to know the value of K up to a constant multiplicative factor, and there are at most polynomially many values to choose from (e.g., guess the heaviest vertex v in A^*, and guess which value of $1 \leq i \leq \log n$ makes $2^i\pi(v)$ the closest estimate for $\pi(A^*)$). Alternatively, the assumption can be dropped at the expense of a more complicated relaxation.

Spreading constraint. The following constraint expresses our guess for the value of K.

$$\tfrac{1}{2} \sum_{i,j \in V} \pi(i)\pi(j)(x_i - x_j)^2 = K.$$

Notice that $(x_i - x_j)^2 = 1$ if and only if $\{x_i, x_j\} = \{0,1\}$.

The objective function. Finally, we write down the objective we are trying to minimize:

$$\text{minimize} \qquad \frac{1}{K}\sum_{i\in V}\pi(i)s_i^2.$$

The above quadratic program computes exactly the value of $\tilde{\alpha}_\pi(G)$, and hence is NP-hard to solve.

9.3.2 The Vector Relaxation

We relax the quadratic program of Section 9.3.1 to a "vector" program that can be solved up to arbitrary precision in polynomial time. The relaxation involves two aspects.

Interpretation of variables. All variables are allowed to be arbitrary vectors in $\mathbb{R}^d$, rather than in $\mathbb{R}$. The dimension d is not constrained, and might be as large as the number of variables (i.e., $3n$).

Interpretation of products. The original quadratic program involved products over pairs of variables. Every such product is interpreted as an inner product between the respective vector variables. The exclusion constraints merely force vectors to be orthogonal (rather than forcing one of them to be 0), and the integrality constraints are no longer implied by the exclusion and choice constraints. The choice constraints imply (among other things) that no vector has norm greater than 1, and the edge constraints imply that whenever $(i, j) \in E$, the corresponding vectors x_i and y_j are orthogonal.

9.3.3 Adding Valid Constraints

We now strengthen the vector program by adding more valid constraints. This should be done in a way that will not violate feasibility (in cases where the original quadratic program was feasible), and moreover, that preserves polynomial time solvability (up to arbitrary precision) of the resulting vector program. It is known that this last condition is satisfied if we only add constraints that are linear over inner products of pairs of vectors, and this is indeed what we shall do. The reader is encouraged to check that every constraint that we add is indeed satisfied by feasible 0/1 solutions to the original quadratic program.

The 1-vector. We add additional variable v to the vector program. It is our intention that variables whose value is 1 in the quadratic program will have value equal to that of v in the vector program. Hence v is a unit vector, and we add the constraint $v^2 = 1$.

Sphere constraints. For every vector variable z we add the constraint $z^2 = vz$. Geometrically, this forces all vectors to lie on the surface of a sphere of radius $\frac{1}{2}$ centered at $\frac{v}{2}$ because the constraint is equivalent to $(z - \frac{v}{2})^2 = \frac{1}{4}$.

Triangle constraints. For every three variables z_1, z_2, z_3 we add the constraint

$$(z_1 - z_2)^2 + (z_2 - z_3)^2 \geq (z_1 - z_3)^2.$$

This implies that every three variables (which are points on the sphere $S(\frac{v}{2}, \frac{1}{2})$) form a triangle whose angles are all at most $\pi/2$.

Removing the s_i vectors. In the upcoming sections we shall describe and analyze a rounding procedure for our vector program. It turns out that our rounding procedure does not use the vectors s_i, only the values $s_i^2 = 1 - x_i^2 - y_i^2$. Hence we can modify the choice constraints to

$$x_i^2 + y_i^2 \leq 1$$

and remove all explicit mention of the s_i vectors, without affecting our analysis for the rounding procedure. Similarly, it suffices to include in the vector program only those triangle constraints in which all three vectors are x vectors. The full vector program follows.

$$
\begin{array}{lll}
\text{minimize} & \frac{1}{K}\sum_{i\in V}\pi(i)(1-x_i^2-y_i^2) & \\
& & \\
\text{subject to} & x_i, y_i, v \in \mathbb{R}^{2n}, & i \in V \\
& x_i^2 + y_i^2 \le 1, & i \in V \\
& x_i \cdot y_i = 0, & i \in V \\
& x_i \cdot y_j = x_j \cdot y_i = 0, & (i,j) \in E \\
& v^2 = 1 & \\
& v \cdot x_i = x_i^2,\ v \cdot y_i = y_i^2, & i \in V \\
& \frac{1}{2}\sum_{i,j\in V}\pi(i)\pi(j)(x_i - x_j)^2 = K & \\
& (x_i - x_j)^2 \le (x_i - x_h)^2 + (x_h - x_j)^2, & h, i, j \in V.
\end{array}
$$

In the following section, we will show how to round this to a solution which is within an $O(\sqrt{\log k})$ factor of optimal. In Section 9.6, we show that this analysis is tight, even for a family of stronger (i.e. more constrained) vector programs.

9.4 Algorithmic Framework for Rounding

In this section, we develop a general algorithmic framework for rounding solutions to optimization problems on vertex cuts. We begin with a classical theorem.

Theorem 9.1 (Menger's theorem). *A graph $G = (V, E)$ contains at least k vertex-disjoint paths between two non-adjacent vertices $u, v \in V$ if and only if every vertex cut that separates u from v has size at least k.*

It is well-known that a smallest vertex cut separating u from v can be found in polynomial time (in the size of G) by deriving Menger's Theorem from the Max-Flow-Min-Cut Theorem (see e.g. [165]).

Suppose that, in addition to a graph $G = (V, E)$, we have a set of non-negative vertex capacities $\{c_v\}_{v\in V} \subseteq \mathbb{N}$. (For simplicity, we are assuming here that capacities are integer, but the following discussion can also be extended to the case of arbitrary nonnegative capacities.) For a subset $S \subseteq V$, we define $\mathrm{cap}(S) = \sum_{v \in S} c_v$. We have the following immediate corollary whose proof is deferred.

Corollary 9.2. *Let $G = (V, E)$ be a graph with vertex capacities. Then for any two non-adjacent vertices $u, v \in V$, the following two statements are equivalent.*

1. *Every vertex cut $S \subseteq V$ that separates u from v has $\mathrm{cap}(S) \geq k$.*

2. *There exist u-v paths $p_1, p_2, \ldots, p_k \subseteq V$ such that for every $w \in V$,*

$$\#\{1 \leq i \leq k : w \in p_i\} \leq c_w.$$

Furthermore, a vertex cut S of minimal capacity can be found in polynomial time.

Proof. The proof is by a simple reduction. From $G = (V, E)$ and the capacities $\{c_v\}_{v \in V}$, we create a new uncapacitated instance $G' = (V', E')$ and then apply Menger's theorem to G'.

To arrive at G', we replace every vertex $v \in V$ with a collection of representatives $v_1, v_2, \ldots, v_{c_v}$ (if $c_v = 0$, then this corresponds to deleting v from the graph). Now for every edge $(u, v) \in E$, we add edges $\{(u_i, v_j) : 1 \leq i \leq c_u, 1 \leq j \leq c_v\}$. It is not hard to see that every minimal vertex cut either takes all representatives of a vertex or none, giving a one-to-one correspondence between minimal vertex cuts in G and G'. □

Furthermore, given such a capacitated instance $G = (V, E), \{c_v\}_{v \in V}$ along with $u, v \in V$, it is possible to find, in polynomial time, a vertex cut $S \subseteq V$ of minimal capacity which separates u from v.

Suppose additionally that we have a *demand function* $\omega : V \times V \to \mathbb{R}_+$ which is *symmetric*, i.e. $\omega(u, v) = \omega(v, u)$. In this case, we define the *sparsity of* (A, B, S) *with respect to* ω by

$$\alpha^{\mathrm{cap},\omega}(A, B, S) = \frac{\mathrm{cap}(S)}{\sum_{u \in A \cup S} \sum_{v \in B \cup S} \omega(u, v)}.$$

We define the *sparsity of* G by $\alpha^{\text{cap},\omega}(G) = \min\{\alpha^{\text{cap},\omega}(A, B, S)\}$ where the minimum is taken over all vertex separators. Note that $\alpha_\pi(A, B, S) = \alpha^{\text{cap},\omega}(A, B, S)$ when $c_v = \pi(v)$ and $\omega(u, v) = \pi(u)\pi(v)$ for all $u, v \in V$.

9.4.1 Line Embeddings and Vertex Separators

Let $G = (V, E)$ be a graph with vertex capacities $\{c_v\}_{v \in V}$, and a demand function $\omega : V \times V \to \mathbb{R}_+$. Furthermore, suppose that we have a map $f : V \to \mathbb{R}$. We give the following algorithm which computes a vertex cut (A, B, S) in G.

Algorithm FINDCUT(G, f)

1. Sort the vertices $v \in V$ according to the value of $f(v)$: $\{y_1, y_2, \ldots, y_n\}$.
2. For each $1 \leq i \leq n$,
3. Create the augmented graph $G_i = (V \cup \{s, t\}, E_i)$ with
   $E_i = E \cup \{(s, y_j), (y_k, t) : 1 \leq j \leq i, i < k \leq n\}$.
4. Find the minimum capacity s-t separator S_i in G_i.
5. Let $A_i \cup \{s\}$ be the component of $G[V \cup \{s, t\} \setminus S_i]$ containing s,
   let $B_i = V \setminus (A_i \cup S_i)$.
6. Output the vertex separator (A_i, B_i, S_i) for which $\alpha^{\text{cap},\omega}(A_i, B_i, S_i)$ is minimal.

The analysis. Suppose that we have a cost function $\mathsf{cost} : V \to \mathbb{R}_+$. We say that the map $f : V \to \mathbb{R}$ is *path-compatible with the cost function* cost if, for any path $p = v_1, v_2, \ldots, v_k$ in G,

$$\sum_{i=1}^{k} \mathsf{cost}(v_i) \geq |f(v_1) - f(v_k)|. \tag{9.1}$$

We now move onto the main lemma of this section.

Lemma 9.3 (Charging lemma). *Let $G = (V, E)$ be any capacitated graph with demand function $\omega : V \times V \to \mathbb{R}_+$. Suppose additionally that we have a cost function* cost : *$V \to \mathbb{R}_+$ and a path-compatible map $f : V \to \mathbb{R}$. If α_0 is the sparsity of the minimum* *ratio vertex cut output by* FINDCUT(G, f)*, then*

$$\sum_{v \in V} c_v \cdot \mathsf{cost}(v) \geq \frac{\alpha_0}{2} \sum_{u,v \in V} \omega(u,v)|f(u) - f(v)|.$$

Proof. Recall that we have sorted the vertices v according to the value of $f(v)$: $\{y_1, y_2, \ldots, y_n\}$. Let $C_i = \{y_1, \ldots, y_i\}$ and $\varepsilon_i = f(y_{i+1}) - f(y_i)$. First we have the following lemma which relates the size of the separators found to the average distance under f, according to ω.

Lemma 9.4.

$$\sum_{i=1}^{n-1} \varepsilon_i \operatorname{cap}(S_i) \geq \alpha_0 \sum_{u,v \in V} \omega(u,v)|f(u) - f(v)|.$$

Proof. Using the fact that α_0 is the minimum sparsity of all cuts found by FINDCUT(G, f),

$$\begin{aligned} \operatorname{cap}(S_i) &\geq \alpha_0 \sum_{u \in A_i \cup S_i} \sum_{v \in B_i \cup S_i} \omega(u,v) \\ &\geq \alpha_0 \sum_{u \in C_i} \sum_{v \in V \setminus C_i} \omega(u,v). \end{aligned}$$

Multiplying both sides of the previous inequality by ε_i and summing over $i \in \{1, 2, \ldots, n-1\}$ proves the lemma. □

Now we come to the heart of the charging argument which relates the cost function to the capacity of the cuts occurring in the algorithm.

Lemma 9.5 (Charging against balls).

$$\sum_{v \in V} c_v \cdot \mathsf{cost}(v) \geq \frac{1}{2} \sum_{i=1}^{n-1} \varepsilon_i \operatorname{cap}(S_i).$$

Proof. We show how to charge every "unit" of $\sum_{i=1}^{n-1} \varepsilon_i \operatorname{cap}(S_i)$ against vertices $v \in V$ so that no vertex is charged more than $2\, c_v \cdot \mathsf{cost}(v)$. This is done as follows. For every vertex $v \in V$, we surround $f(v) \in \mathbb{R}$ by a closed ball of radius $\mathsf{cost}(v)$. This ball has width $2\,\mathsf{cost}(v)$ on the line. We will be charging various amounts to segments of this width, never charging a segment of v's ball more than c_v.

Fix some $1 \leq i \leq n-1$. Since S_i is a minimum capacity s-t separator, applying Menger's theorem yields a family of $m = |S_i|$ s-t paths $p_1, \ldots, p_m$ which use no vertex $v \in V$ more than c_v times. Since these paths cross from C_i to $V \setminus C_i$, there must exist subpaths $q_1, \ldots, q_m$ for which each initial vertex is in C_i and each final vertex is in $V \setminus C_i$.

For each path q_j, we will charge ε_i against $\sum_{v \in V} c_v \cdot \mathsf{cost}(v)$. Let $q_j = u_1, u_2, \ldots, u_r$. First, we note that since f is path-compatible with cost, the balls $\{B(u_i, \mathsf{cost}(u_i))\}_{i=1}^r$ must cover the entire interval $[f(u_1), f(u_r)]$. Furthermore, we have $f(u_r) - f(u_1) \geq \varepsilon_i$ by construction.

To charge ε_i to the balls belonging to points in the path q_j, we proceed as follows. Every point z of the interval $[f(u_1), f(u_r)]$ is contained in at least one ball $B(u_i, \mathsf{cost}(u_i))$. We will charge this point against the corresponding ball, in fact we charge it to $z \in B(u_i, \mathsf{cost}(u_i))$. (It may be the case that many of the balls contain z. We can just charge all of them.) Clearly this charges all of ε_i.

We are only left to see that every vertex is charged at most $2\,c_v \cdot \mathsf{cost}(v)$. But this is easy; note that a point $z \in B(v, \mathsf{cost}(v))$ can only be charged in the round corresponding to the segment of distance ε_i in which z is contained. Secondly, notice that in this round it can only be charged c_v times since v occurs in at most c_v of the paths $q_1, \ldots, q_j$. This completes the proof of Lemma 9.5. □

Combining Lemmas 9.4 and 9.5 finishes the proof of Lemma 9.3. □

9.4.2 Line Embeddings and Distortion

Let (X, d) be a metric space. A map $f : X \to \mathbb{R}$ is called *1-Lipschitz* if, for all $x, y \in X$,

$$|f(x) - f(y)| \leq d(x, y).$$

Given a 1-Lipschitz map f and a demand function $\omega : X \times X \to \mathbb{R}_+$, we define its *average distortion under* ω by

$$avd_\omega(f) = \frac{\sum_{x,y \in X} \omega(x, y) \cdot d(x, y)}{\sum_{x,y \in X} \omega(x, y) \cdot |f(x) - f(y)|}.$$

We say that a weight function ω is a *product weight* if it can be written as $\omega(x, y) = \pi(x)\pi(y)$ for all $x, y \in X$, for some $\pi : X \to \mathbb{R}_+$. We now state three theorems which give line embeddings of small average distortion in various settings. The proofs of these theorems are described in [90].

Theorem 9.6 (Bourgain, [43]). *If (X, d) is an n-point metric space, then for every weight function $\omega : X \times X \to \mathbb{R}_+$, there exists an efficiently computable map $f : X \to \mathbb{R}$ with $avd_\omega(f) = O(\log n)$.*

Theorem 9.7 (Rabinovich, [139]). *If (X, d) is any metric space supported on a graph which excludes a K_r-minor, then for every* product weight *$\omega_0 : X \times X \to \mathbb{R}_+$, there exists an efficiently computable map $f : X \to \mathbb{R}$ with $avd_{\omega_0}(f) = O(r^2)$.*

Theorem 9.8 (Arora, Rao, Vazirani, [19]). *If (X, d) is an n-point metric of negative type, then for every* product weight *$\omega_0 : X \times X \to \mathbb{R}_+$, there exists an efficiently computable map $f : X \to \mathbb{R}$ with $avd_{\omega_0}(f) = O(\sqrt{\log n})$.*

We also recall the following classical result.

Lemma 9.9. *Let (Y, d) be any metric space and $X \subseteq Y$. Given a 1-Lipschitz map $f : X \to \mathbb{R}$, there exists a 1-Lipschitz extension $\tilde{f} : Y \to \mathbb{R}$, i.e. such that $\tilde{f}(x) = f(x)$ for all $x \in X$.*

Proof. One defines

$$\tilde{f}(y) = \sup_{x \in X} [f(x) - d(x, y)]$$

for all $y \in Y$. □

9.4.3 Analysis of the Vector Program

We now continue our analysis of the vector program from Section 9.3.3. Recall that $\pi(i)(1 - x_i^2 - y_i^2)$ is the contribution of vertex i to the objective function. For every $i \in V$, define $\mathsf{cost}(i) = 4(1 - x_i^2 - y_i^2)$. We will consider the metric space (V, d) given by $d(i, j) = (x_i - x_j)^2$ (note that this is a metric space precisely because every valid solution to the SDP must satisfy the triangle inequality constraints). The following key proposition allows us to apply the techniques of Sections 9.4.1 and 9.4.2 to the solution of the vector program.

Proposition 9.10. *For every edge $(i,j) \in E$, $\mathsf{cost}(i) + \mathsf{cost}(j) \geq 2(x_i - x_j)^2$.*

Proof. Since $(i,j) \in E$, we have $x_i \cdot y_j = x_j \cdot y_i = 0$, and recall that $x_i \cdot y_i = x_j \cdot y_j = 0$. It follows that

$$(x_i - x_j)^2 \leq 2[(x_i + y_i - v)^2 + (x_j + y_i - v)^2] \leq 2[(1 - x_i^2 - y_i^2) + (1 - x_j^2 - y_i^2)].$$

(The first inequality follows from the fact that $(x_i - x_j)^2 = ((x_i + y_i - v) - (x_j + y_i - v))^2$, and from the fact that $(x+y)^2 \geq 0$, implying $x^2 + y^2 \geq 2xy$ and $(x-y)^2 \leq 2(x^2+y^2)$. Substitute $x = x_i + y_i - v$ and $y = x_j + y_i - v$.)

Putting y_j instead of y_i in the above equation gives $(x_i - x_j)^2 \leq 2[(1 - x_i^2 - y_j^2) + (1 - x_j^2 - y_j^2)]$. Summing these two inequalities yields

$$2(x_i - x_j)^2 \leq 4[(1 - x_i^2 - y_i^2) + (1 - x_j^2 - y_j^2)] \leq \mathsf{cost}(i) + \mathsf{cost}(j).$$

□

Now, let $U = \mathrm{supp}(\pi) = \{i \in V : \pi(i) \neq 0\}$, and put $k = |U|$. Finally, let $f : (U, d) \to \mathbb{R}$ be any 1-Lipschitz map, and let $\tilde{f} : V \to \mathbb{R}$ be the 1-Lipschitz extension guaranteed by Lemma 9.9.

Then for any path $v_1, \ldots, v_m$ in G, we have

$$\begin{aligned} |\tilde{f}(v_1) - \tilde{f}(v_m)| \leq d(v_1, v_m) \leq \sum_{i=1}^{m-1} d(v_i, v_{i+1}) &= \sum_{i=1}^{m} (x_{v_i} - x_{v_{i+1}})^2 \\ &\leq \frac{1}{2} \sum_{i=1}^{m} (\mathsf{cost}(v_i) + \mathsf{cost}(v_{i+1})) \\ &\leq \sum_{i=1}^{m} \mathsf{cost}(v_i). \end{aligned}$$

where the third inequality is from Proposition 9.10. We conclude that $\tilde{f}$ is path-compatible with cost.

Defining a product demand by $\omega(i,j) = \pi(i)\pi(j)$ for every $i, j \in V$ and capacities $c_i = \pi(i)$, we now apply $\textsc{FindCut}(G, \tilde{f})$. If the best separator found has sparsity α_0, then by Lemma 9.3,

$$\begin{aligned}
\frac{1}{K}\sum_{i\in V}\pi(i)(1-x_i^2-y_i^2) = \frac{1}{4K}\sum_{i\in V} c_i\cdot\mathsf{cost}(i) \;&\geq\; \frac{\alpha_0}{8K}\sum_{i,j\in V}\omega(i,j)\cdot|\tilde{f}(i)-\tilde{f}(j)| \\
&=\; \frac{\alpha_0}{8K}\sum_{i,j\in U}\omega(i,j)\cdot|f(i)-f(j)| \\
&\geq\; \frac{\alpha_0}{4}\cdot\frac{\sum_{i,j\in U}\omega(i,j)\cdot|f(i)-f(j)|}{\sum_{i,j\in U}\omega(i,j)\cdot d(i,j)} \\
&=\; \frac{\alpha_0}{4\cdot avd_\omega(f)}.
\end{aligned}$$

It follows that $\tilde{\alpha}_\pi(G) \geq \alpha_0/(4\cdot avd_\omega(f))$. Since the metric (V, d) is of negative type and $\omega(\cdot,\cdot)$ is a product weight, we can achieve $avd_\omega(f) = O(\sqrt{\log k})$ using Theorem 9.8. Using this f, it follows that $\textsc{FindCut}(G, \tilde{f})$ returns a separator (A, B, S) such that $\alpha_\pi(A, B, S) \leq O(\sqrt{\log k})\,\tilde{\alpha}_\pi(G)$, completing the analysis.

Theorem 9.11. *Given a graph $G = (V, E)$ and vertex weights $\pi : V \to \mathbb{R}_+$, there exists a polynomial-time algorithm which computes a vertex separator (A, B, S) for which*

$$\alpha_\pi(A, B, S) \leq O(\sqrt{\log k})\,\alpha_\pi(G),$$

where $k = |\mathrm{supp}(\pi)|$.

We extend this theorem to more general weights in Section 9.7.1. This is necessary for some of the applications in Section 9.7.

9.5 Approximate Max-Flow/Min-Vertex-Cut Theorems

Let $G = (V, E)$ be a graph with capacities $\{c_v\}_{v\in V}$ on vertices and a demand function $\omega : V \times V \to \mathbb{R}_+$. The *maximum concurrent vertex flow* of this instance is the maximum constant $\varepsilon \in [0, 1]$ such that one can simultaneously route an ε fraction of each u-v demand $\omega(u,v)$ without violating the capacity constraints.

Let $\mathcal{P}_{uv}$ be the set of all u-v paths in G, and for each $p \in \mathcal{P}_{uv}$, let p^{uv} denote the amount of the u-v commodity that is sent from u to v along p. We now write a linear program that computes the maximum concurrent vertex flow.

$$\begin{array}{lll}
\text{maximize} & \varepsilon & \\
\text{subject to} & \displaystyle\sum_{p\in\mathcal{P}_{uv}} p^{uv} \geq \varepsilon \cdot \omega(u,v), & u,v \in V \\
& \displaystyle\sum_{u,v\in V}\ \sum_{p\in\mathcal{P}_{uv}: w\in p} p^{uv} \leq c_w, & w \in V \\
& p^{uv} \geq 0 & u,v \in V, p \in \mathcal{P}_{uv}
\end{array}$$

We now write the dual of this LP with variables $\{s_v\}_{v\in V}$ and $\{\ell_{uv}\}_{u,v\in V}$.

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{v\in V} c_v s_v & \\
\text{subject to} & \displaystyle\sum_{w\in p} s_w \geq \ell_{uv} & p \in \mathcal{P}_{uv}, \forall u,v \in V \\
& \displaystyle\sum_{u,v\in V} \omega(u,v)\ell_{uv} \geq 1 & \\
& \ell_{uv} \geq 0, s_v \geq 0 & u,v \in V.
\end{array}$$

Finally, define

$$\mathsf{dist}(u,v) = \min_{p\in\mathcal{P}_{uv}} \sum_{w\in p} s_w.$$

By setting $\ell_{uv} = \mathsf{dist}(u,v)$, we see that the above dual LP is equivalent to the following.

$$\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{v\in V} c_v s_v \\
\text{subject to} & \displaystyle\sum_{u,v} \omega(u,v) \cdot \mathsf{dist}(u,v) \geq 1.
\end{array}$$

Remark. The dual LP above is derived by defining the length of a path $p = v_1, \ldots, v_k$

to be $\sum_{1\le i\le k} s_{v_i}$. However, a "cleaner" definition would take only half of the contribution of the endpoints of the path, giving it length $s_{v_1}/2 + \sum_{2\le i\le k-1} s_{v_i} + s_{v_k}/2$. Under this second definition, each edge (u,v) has length $(s_u + s_v)/2$, and the length of a path is the sum of the lengths of its respective edges. In terms of approximating vertex cuts, the two definitions differ by a factor of at most two, which makes little difference for us. We choose to work with the first definition because it simplifies some of the notation. However, in Section 9.5.3 we shall switch to the second definition.

9.5.1 Rounding to Vertex Separators

Observe that any vertex separator (A, B, S) yields an upper bound on the maximum concurrent flow in G. The upper bound is of the form

$$\alpha^{\text{cap},\omega}(A,B,S) = \frac{\text{cap}(S)}{\sum_{u\in A\cup S}\sum_{v\in B\cup S}\omega(u,v)}.$$

The numerator is the capacity of the separator, while the denominator is the amount of demand that must be sent through it. To see how tight this upper bound is in general, we can take the dual of the max-concurrent-flow LP from the previous section and round it to a vertex separator while increasing the cost by at most some factor.

We will write $\alpha = \alpha^{\text{cap},\omega}$ if the capacity and demands are clear from context. We note that the dual LP is a relaxation of $\alpha(G)$, since every vertex separator (A, B, S) gives a feasible solution, where $s_v = 1/\lambda$ if $v \in S$ and $s_v = 0$ otherwise. In this case $\mathsf{dist}(u,v) \ge 1/\lambda$ if $u \in A\cup S$ and $v \in B \cup S$ or visa-versa, so that setting $\lambda = \sum_{u\in A\cup S, v\in B\cup S}\omega(u,v)$ yields a feasible solution.

9.5.2 The Rounding

Before presenting our approach for rounding the LP, let us recall a typical rounding approach for the case of *edge-capacitated* flows. In the edge context [131, 21], one observes that the dual LP is essentially integral when $\mathsf{dist}(\cdot,\cdot)$ forms an L_1 metric. To round in the case when $\mathsf{dist}(\cdot,\cdot)$ does not form an L_1 metric, one uses Bourgain's theorem [43] to embed (V, dist) into L_1 (with $O(\log n)$ distortion, that translates to a similar loss in the approximation ratio), and then rounds the resulting L_1 metric (where rounding the L_1 metric does not incur a loss in the approximation ratio). This approach is not as effective in the case of vertex separators, because rounding an L_1 metric does incur a loss in the approximation ratio (as the example below shows), and hence there is not much point in embedding (V, dist) into L_1 and paying the distortion factor.

The discrete cube. Let $G = (V, E)$ be the d-dimensional discrete hypercube $\{0,1\}^d$. We put $c_v = 1$ for every $v \in V$, and $\omega(u, v) = 1$ for every pair $u, v \in V$. It is well-known that $\alpha^{\mathrm{cap},\omega}(G) = \Theta(1/(2^d\sqrt{d}))$ [111]. On the other hand, consider the fractional separator (i.e. dual solution) given by $s_v = 10 \cdot \frac{4^{-d}}{d}$. Note that $\mathsf{dist}(u, v)$ is proportional to the shortest-path metric on the standard cube, hence $\sum_{u,v} \mathsf{dist}(u, v) \geq 1$, yielding a feasible solution which is a factor $\Theta(\sqrt{d})$ away from $\alpha(G)$.

It follows that even when (V, dist) is an L_1 metric, the integrality gap of the dual LP might be as large as $\Omega(\sqrt{\log n})$.

Rounding with line embeddings. The rounding is done as follows. Let $\{s_v\}_{v\in V}$ be an optimal solution to the dual LP, and let $\mathsf{dist}(\cdot,\cdot)$ be the corresponding metric on V. Suppose that the demand function $\omega : V \times V \to \mathbb{R}_+$ is supported on a set S, i.e. $\omega(u, v) > 0$ only if $u, v \in S$, and that $|S| = k$. Let $f : (S, \mathsf{dist}) \to \mathbb{R}$ be the map guaranteed by Theorem 9.6 with $avd_\omega(f) = O(\log k)$, and let $\tilde{f} : (V, \mathsf{dist}) \to \mathbb{R}$ be the 1-Lipschitz extension from Lemma 9.9.

For $v \in V$, define $\mathsf{cost}(v) = s_v$. Then since $\tilde{f}$ is 1-Lipschitz, for a path $v_1, v_2, \ldots, v_m$ in G, we have

$$\sum_{i=1}^{m} \mathsf{cost}(v_i) \geq \mathsf{dist}(v_1, v_m) \geq |\tilde{f}(v_1) - \tilde{f}(v_m)|,$$

hence $\tilde{f}$ is path-compatible with cost.

We now apply $\textsc{FindCut}(G, \tilde{f})$. If the best separator found has sparsity α_0, then by Lemma 9.3,

$$\begin{aligned}\sum_v c_v s_v = \sum_v c_v \cdot \mathsf{cost}(v) &\geq \alpha_0 \sum_{u,v \in V} \omega(u,v)\,|\tilde{f}(u) - \tilde{f}(v)| \\ &= \alpha_0 \sum_{u,v \in S} \omega(u,v)\,|f(u) - f(v)| \\ &\geq \Omega\left(\frac{\alpha_0}{\log k}\right) \sum_{u,v \in V} \omega(u,v)\,\mathsf{dist}(u,v) \geq \Omega\left(\frac{\alpha_0}{\log k}\right).\end{aligned}$$

Theorem 9.12. *For an arbitrary vertex-capacitated flow instance, where the demand is supported on a set of size k, there is an $O(\log k)$-approximate max-flow/min-vertex-cut theorem. In particular, this holds if there are only k commodities.*

9.5.3 Excluded Minor Families

Here we shall switch to the "cleaner" definition for path lengths that is presented in the remark following the description of the dual LP. This allows us to view path lengths as being induced by edge lengths. A consequence of this is that if the graph G excludes some fixed graph H as a minor, then the metric arising from the LP dual is an H-excluded metric. It follows that applying Theorem 9.7, yields a better result when G excludes a minor and when we have a product demand function $\omega(u,v)$.

Theorem 9.13. *When G is an H-minor-free graph, there is an $O(|V(H)|^2)$-approximate max-flow/min-vertex-cut theorem with product demands. Additionally, there exists an $O(|V(H)|^2)$ approximation algorithm for finding min-quotient vertex cuts in G.*

9.6 An Integrality Gap for the Vector Program

Consider the hypercube graph. Namely, the n vertices of the graph (where n is a power of 2) can be viewed as all vectors in $\{\pm 1\}^{\log n}$, and edges connect two vertices that differ in exactly one coordinate. Every vertex separator (A, B, S) has $\alpha(A, B, S) \geq 1/O(n\sqrt{\log n})$. This follows from standard vertex isoperimetry on the cube [111]. We show a solution to the vector program with value of $O(n/\log n)$, proving an integrality ratio of $\Omega(\sqrt{\log n})$ for the vector program, and implying that our rounding technique achieves the best possible approximation ratio (relative to the vector program), up to constant multiplicative factors.

In the solution to the vector program, we describe for every vertex i the associated vectors x_i and y_i. The vectors s_i will not be described explicitly, but are implicit, using the relation $s_i = v - x_i - y_i$. Each vector will be described as a vector in $1 + n \log n + 2(n-1)$ dimensions (even though n dimensions certainly suffice). Our redundant representation in terms of number of dimensions helps clarify the structure of the solution.

To describe the vector solution, we introduce two parameters, a and b. Their exact value will be determined later, and will turn out to be $a = 1/2 - \Theta(1/\log n)$, $b = \Theta(1/\sqrt{n \log n})$. We partition the coordinates into three groups of coordinates.

G1. Group 1, containing one coordinate. This coordinate corresponds to the direction of vector v (which has value 1 in this coordinate and 0 elsewhere). All x_i and y_i vectors have value a on this coordinate.

G2. Group 2, containing n identical blocks of $\log n$ coordinates. The coordinates within a block exactly correspond to the structure of the hypercube. Within a block, each x_i is a vector in $\{\pm b\}^{\log n}$ derived by scaling the hypercube label of vertex i (which is a vector in $\{\pm 1\}^{\log n}$) by a factor of b. Vector y_i is the negation of vector x_i on the coordinates of Group 2.

G3. Group 3, containing 2 identical blocks of $n-1$ coordinates. The coordinates within a block arrange all the x_i vectors as vertices of a simplex. This is done in the following way. Let H_n be the n by n Hadamard matrix with entries ± 1, obtained by taking the $(\log n)$-fold tensor product of the 2 by 2 the matrix H_2 that has rows $(1,1)$ and $(1,-1)$. The inner product of any two rows of H_n is 0, the first column is all 1, and the sum of entries in any other column is 0. Remove the first column to obtain the matrix H'_n. Within a block, let vector x_i be the ith row of H'_n, scaled by a factor of b. Hence within a block, $x_i x_i = b^2(n-1)$, and $x_i x_j = -b^2$ for $i \neq j$. Vector y_i is identical to x_i on the coordinates of Group 3.

We now show that the triangle constraints are satisfied by our vector solution. Recall (see Section 9.3) that there is some flexibility in the choice of which triangle constraints to include in the vector program (and likewise for many other constraints that are valid for 0/1 solutions but are not used in our analysis). We shall address here a subset of the triangle constraints that is larger than that actually used in the analysis of our rounding algorithm.

There are five sets of vectors from which we can take the three vectors that participate in a triangle constraint: X (the x_i vectors), Y (the y_i vectors), S (the s_i vectors), v and 0. In our analysis we used only triangle constraints over vectors from X. Here we show that all the triangle constraints that involve only vectors from $X \bigcup Y$ are satisfied. All vectors in $X \bigcup Y$ have the identical value a in their first coordinate, and in every other coordinate they take only values from $\pm b$. Hence per coordinate, *every quadratic constraint that holds for all* ± 1 *vectors* (including, but not limited to, the triangle constraints) is satisfied for all x_i and y_i vectors.

We let $K = \sum_{i,j \in V}(x_i - x_j)^2 = \Theta(n^3 b^2 \log n)$. The value of the parameters a and b is governed by the following three constraints.

1. The exclusion constraints imply that

$$a^2 - nb^2 \log n + 2b^2(n-1) = 0$$

2. The edge constraints (and the fact that edges connect vertices of Hamming distance 1) imply that

$$a^2 - nb^2(\log n - 2) - 2b^2 = 0$$

3. The sphere constraints imply that

$$a = a^2 + nb^2 \log n + 2b^2(n-1)$$

Hence we have a system of three equalities in two unknowns (a and b). This system is consistent, because the first two equalities are in fact identical (due to our careful choice of number of blocks in each group). They both give:

$$a^2 + (-n\log n + 2n - 2)b^2 = 0$$

By setting $b = a/\sqrt{n\log n - 2n + 2}$ the first two equalities are satisfied. The third equality now reads $a = a^2(2+\varepsilon)$ for some $\varepsilon = \Theta(1/\log n)$. This equality is satisfied by taking a roughly equal to $1/2 - \varepsilon/4$, which is $1/2 - \Theta(1/\log n)$.

It follows that in the vector solution all $s_i^2 = 1 - x_i^2 - y_i^2$ is $O(1/\log n)$ for every $i \in V$. Hence our vector solution has value

$$\frac{1}{K}\sum_{i\in V} s_i^2 = \frac{1}{\Theta(n\log n)}.$$

Finally we note that rather than have only one coordinate in Group 1, we can have $(a/b)^2 = n\log n - 2n + 2$ coordinates, and give the x and y vectors values b in these coordinates. Then all x and y vectors become vertices of a $2n\log n$-dimensional hypercube (of side length b). We see that even in this special case, the integrality gap remains $\Omega(\sqrt{\log n})$.

9.7 Balanced Vertex Separators and Applications

9.7.1 More General Weights

An important generalization of the min-ratio vertex cut introduced in Section 9.3 is when a pair of weight functions $\pi_1, \pi_2 : V \to \mathbb{R}_+$ is given and one wants to find the vertex separator (A, B, S) which minimizes

$$\alpha_{\pi_1,\pi_2}(A,B,S) = \frac{\pi_1(S)}{\pi_2(A\cup S)\cdot \pi_2(B\cup S)}.$$

We can also give an $O(\sqrt{\log k})$ approximation in this case, where again $k = |\mathrm{supp}(\pi_2)|$.

Let

$$\tilde{\alpha}_{\pi_1,\pi_2}(A,B,S) = \pi_1(S)/[\pi_2(A)\cdot\pi_2(B\cup S)],$$

where $\pi_2(A) \geq \pi_2(B)$. Also define $\alpha_{\pi_1,\pi_2}(G)$ and $\tilde{\alpha}_{\pi_1,\pi_2}(G)$ as before. Note that by changing the vector program to minimize $\frac{1}{K}\sum_{i\in V}\pi_1(i)(1-x_i^2-y_i^2)$, it becomes a relaxation for $\tilde{\alpha}_{\pi_1,\pi_2}(G)$. Similarly, the rounding analysis goes through unchanged to yield a separator (A,B,S) with

$$\alpha_{\pi_1,\pi_2}(A,B,S) \leq O(\sqrt{\log k})\,\tilde{\alpha}_{\pi_1,\pi_2}(G).$$

The only difficulty is that if (A^*,B^*,S^*) is the optimal separator, the relation between the two notions, $\alpha_{\pi_1,\pi_2}(A^*,B^*,S^*)$ and $\tilde{\alpha}_{\pi_1,\pi_2}(A^*,B^*,S^*)$, is no longer as clear. If they are not within a factor of 2, then it must hold that $\pi_2(A^*\cup S^*) \geq 2\,\pi_2(A^*)$, i.e. $\pi_2(S^*) \geq \pi_2(A^*)$. (Where we assume that $\pi_2(A^*) \geq \pi_2(B^*)$.)

In this case,

$$\frac{\pi_1(S^*)}{\pi_2(S^*)^2} \leq 4\,\alpha_{\pi_1,\pi_2}(G).$$

Hence it suffices to find an approximation for a different problem, that of finding a subset $S \subseteq V$ which minimizes the ratio $\pi_1(S)/\pi_2(S)^2$. This problem can be solved in polynomial time (see e.g. [90]).

Theorem 9.14. *Given a graph $G=(V,E)$ and vertex weights $\pi_1,\pi_2 : V \to \mathbb{R}_+$, there exists a polynomial-time algorithm which computes a vertex separator (A,B,S) for which*

$$\alpha_{\pi_1,\pi_2}(A,B,S) \leq O(\sqrt{\log k})\,\alpha_{\pi_1,\pi_2}(G),$$

where $k = |\mathrm{supp}(\pi_2)|$.

We also note that Theorem 9.14 with general weights π_1,π_2 is useful for certain hypergraph partitioning problems [129].

9.7.2 Reduction from Min-Ratio Cuts to Balanced Separators

In this section, we sketch a pseudo-approximation for finding balanced vertex separators in a graph $G = (V, E)$. Let $W \subseteq V$ be an arbitrary subset of V. For $\delta \in (0, 1)$, we say that a subset $X \subseteq V$ is a *δ-vertex-separator (with respect to W)* if every connected component C of $G[V \setminus X]$ has $|C \cap W| \leq \delta |W|$. Our goal in this section is to show that we can find a $\frac{2}{3}$-vertex-separator $X \subseteq V$ whose size is within an $O(\beta)$ factor of the optimal $\frac{1}{2}$-vertex-separator of G, whenever we can find approximate min-ratio cuts in G within factor β. This technique is standard (see [129]).

The algorithm. Let $m = |W|$, and for any subset $U \subseteq V$, define $|U|_W = |U \cap W|$. Let $\pi_1(v) = 1$ for every $v \in V$, and $\pi_2(v) = 1$ if $v \in W$ and $\pi_2(v) = 0$ otherwise. These are the weights for the numerator and denominator, respectively, i.e. we assume that we have a β-approximation for $\alpha_{\pi_1,\pi_2}(\cdot)$. We maintain a vertex separator $S \subseteq V$. Initially, $S = \emptyset$. As long as there exists some connected component $U \subseteq V$ in $G[V \setminus S]$ with $|U|_W \geq \frac{2}{3}|W|$, we use our β-approximation to find a minimum-ratio vertex cut S' in $G[U]$ which is within β of optimal. We then set $S \leftarrow S \cup S'$ and continue.

The analysis. Let S be the final vertex separator. By construction, it is a $\frac{2}{3}$-vertex separator since every connected component U of $G[V \setminus S]$ has $|U|_W < \frac{2}{3}|W|$. Let $T \subseteq V$ be an optimal $\frac{1}{2}$-vertex separator.

Claim 9.15. $|S| \leq O(\beta)|T|$.

Proof. We know that T separates G into two pieces, call them $A_T, B_T \subseteq V$ such that $|A_T \cup T|_W, |B_T \cup T|_W \geq \frac{1}{2}|W|$. Suppose we are at a step where $|U|_W \geq \frac{2}{3}|W|$. Let (A', B', S') be the vertex separator in $G[U]$ that we find by running our min-quotient cut algorithm with ratio β, and suppose that $|A'|_W \geq |B'|_W$. We know that

$$\frac{|S'|}{|A' \cup S'|_W \cdot |B' \cup S'|_W} \leq \beta \frac{|T|}{|(A_T \cup T) \cap U|_W \cdot |(B_T \cup T) \cap U|_W} \leq \frac{12\beta|T|}{m^2},$$

where the final inequality follows because $|U|_w \geq \frac{2m}{3}$. It follows that

$$|S'| \leq \frac{12\beta|T|(|B'|_w + |S'|_w)}{m}.$$

To see that $|S| \leq O(\beta)|T|$, it suffices to see that when we sum $|B'|_w + |S'|_w$ over all iterations, the value is at most $O(m)$. But since we throw away the vertices of $B' \cup S'$ in every iteration (and recurse only on A'), the sum is clearly at most m.

□

9.7.3 Approximating Treewidth

The notion of treewidth has numerous practical applications (see e.g. [33]) and thus a large amount of effort has been put into determining treewidth, which is NP-complete even when the input graph is severely restricted (see the discussion in Section 2.4 for a brief history).

From the approximation viewpoint, Bodlaender et al. [34] gave an $O(\log n)$ approximation algorithm for treewidth on general graphs. Amir [11] improved the approximation factor to $O(\log \mathsf{opt})$ where opt is the actual treewidth of the graph. Constant-factor approximations for treewidth were obtained on AT-free graphs [42, 41] and on planar graphs [155]. The approximation for planar graphs is a consequence of the polynomial-time algorithm given by [155] for computing the parameter *branch-width*, whose value approximates treewidth within a factor of 1.5. Recently, [12] obtained a new approximation algorithm for treewidth in planar graphs with a constant factor slightly worse than 1.5, and the authors of [72] (see also Chapter 2) derived a polynomial-time algorithm for approximating treewidth within a factor of 1.5 for single-crossing-minor-free graphs, generalizations of planar graphs. A well-known open problem is whether treewidth can be approximated within a constant factor. We show that this is indeed the case for every family of graphs that excludes H as a minor, for an arbitrary fixed graph H. For general graphs we improve the approximation factor to $O(\sqrt{\log \mathsf{opt}})$ where opt is the actual treewidth of the graph.

These improvements have several implications, including better approximation algorithms for several other problems like pathwidth, minimum front size, and minimum height elimination tree. They also improve the running time of approximation schemes and fixed-parameter algorithms for several NP-complete problems on graphs of bounded genus, or more generally, graphs excluding a fixed graph as a minor.

Now we are ready to state our approximation result for treewidth.

Theorem 9.16. *There exists a polynomial time algorithm that find a tree decomposition of width at most $O(\sqrt{\log \mathbf{tw}(G)}\ \mathbf{tw}(G))$ for a general graph G and at most $O(|V(H)|^2\ \mathbf{tw}(G))$ for an H-minor-free graph G.*

Proof. The proof is a straightforward modification of the proof of Amir [11]. One recursively uses a balanced vertex-cut algorithm and then proves the aforementioned width bound by induction. In [11] one uses a polynomial-time algorithm that, given a graph $G = (V, E)$ and a set $W \subseteq V$, finds a $\frac{2}{3}$-vertex separator $S \subseteq V$ of W in G of size $O(\log |W|\ k)$, where k is the minimum size of a $\frac{1}{2}$-vertex separator of W in G. Now using Theorems 9.14 and 9.13 and the results of Subsection 9.7.2, we can replace $O(\log |W|\ k)$ with $O(\sqrt{\log |W|}\ k)$ for general graphs and $O(r^2 k)$ for K_r-minor-free graphs, and thus we obtain the desired result. □

By using the result of [34] instead of [11] in the proof of Theorem 9.16, we can obtain a tree decomposition of "logarithmic depth" (in terms of $|V(G)|$) with width at most $O(\sqrt{\log n}\ \mathbf{tw}(G))$ for a general graph G and at most $O(|V(H)|^2\ \mathbf{tw}(G))$ for an H-minor-free graph G.

Improving the approximation factor of treewidth improves the approximation factor for several other problems as follows.

Corollary 9.17. *There exist $O(\sqrt{\log \mathsf{opt}})$ (resp., $O(|V(H)|^2)$) approximation algorithms for branchwidth, minimum front size and minimum size of a clique in a chordal supergraph of a general (resp., H-minor-free) graph G. Additionally, there are $O(\sqrt{\log \mathsf{opt}} \log n)$ (resp., $O(|V(H)|^2 \log n)$) approximation algorithms for pathwidth, minimum height elimination order tree, and search number in a general (resp., H-minor-free) graph G.*

The reader is referred to Bodlaender et al. [34] and Leighton and Rao [129] to see the corresponding exact definitions, references, and the proofs of Corollary 9.17 (the proofs follow almost identical to those of Theorems 17 and 18 of [34] and the fact that treewidth and branchwidth are within a factor 1.5 of each other (see Theorem 2.15).

Improving the approximation factor for treewidth has a direct improvement on the running time of approximation schemes and subexponential fixed-parameter algorithms for several NP-hard problems on graphs families which exclude a fixed minor. In such algorithms finding the tree decomposition of almost minimum width, on which we can run dynamic programming, plays a very important role. More precisely, as mentioned in Chapter 8, Demaine and Hajiaghayi [70, 71] show how one can obtain PTASs for almost all bidimensional parameters on planar graphs, single-crossing-minor-free graphs and bounded-genus graphs. In fact, their approach can be extended to work on apex-minor-free graphs for contraction-bidimensional parameters and on H-minor-free graphs, where H is a fixed graph, for minor-bidimensional parameters. However currently they obtained quasi-polynomial-time approximation schemes for these general settings. The only barrier to obtain PTASs for these general settings is obtaining a constant-factor polynomial-time approximation algorithm for treewidth of an H-minor-free graph for a fixed H (this is posed as an open problem in [70]). Using Theorem 9.16, we overcome this barrier and obtain PTASs for contraction-bidimensional parameters in apex-minor-free graphs and for minor-bidimensional parameters in H-minor-free graphs for a fixed H. As an immediate consequence, we obtain the following theorem (see [70, 71] for the exact definitions of the problems mentioned below).

Theorem 9.18. *There are PTASs for feedback vertex set, vertex cover, minimum maximal matching, and a series of vertex-removal problems in H-minor-free graphs for a fixed H. Also, there are PTASs for dominating set, edge dominating set, r-dominating set, connected dominating set, connected edge dominating set, connected r-dominating set, and clique-transversal set in apex-minor-free graphs.*

Among the problems mentioned above, PTASs for vertex cover and dominating set (but not its other variants) using a different approach were known before (see e.g. [103]).

Chapter 10

Open Problems Regarding Bidimensionality

In this thesis, we introduced the theory of bidimensionality and its applications in algorithmic graph theory. However, still several combinatorial and algorithmic open problems remain in the theory of bidimensionality and related concepts. These problems are usually more general than those mentioned in the end of some previous chapters.

One interesting direction is to generalize bidimensionality to handle general graphs, not just H-minor-free graph classes. As mentioned in Section 1.5, the natural generalization of minor-bidimensionality still yields a parameter-treewidth bound, but it is very large. This direction essentially asks for the size of the largest grid minor guaranteed to exist in any graph of treewidth w. Robertson, Seymour, and Thomas [151] proved that every graph of treewidth larger than 20^{2r^5} has an $r \times r$ grid as a minor, but that some graphs of treewidth $\Omega(r^2 \lg r)$ have no grid larger than $O(r) \times O(r)$, conjecturing that the right requirement on treewidth for an $r \times r$ grid is closer to the $\Theta(r^2 \lg r)$ lower bound. If this conjecture is correct, we would obtain nearly as good parameter-treewidth bounds for minor-bidimensional parameters as in the H-minor-free case. A similar generalization of parameter-treewidth bounds beyond apex-minor-free graphs is not possible for all contraction-bidimensional parameters, e.g., dominating set [62], but it would still be quite interesting to explore an analogous "theory of graph contractions" paralleling the Graph Minor Theory. Such a theory would be an interesting and powerful tool for handling problems that are closed under contractions but not minors, and therefore deserves more focus.

Another interesting direction is to obtain the best constant factors in terms of the fixed excluded minor H. These constants are particularly important in the context of the exponent in the running time of a fixed-parameter algorithm. At the heart of all such constant factors is the lead constant in Theorem 1.6. This factor must be $\Omega(\sqrt{|V(H)|} \lg |V(H)|)$, because otherwise such a bound would contradict the lower bound for general graphs. An upper bound near this lower bound (in particular, polynomial in $|V(H)|$) is not out of the question: the bound on the size of separators in [9] has a lead factor of $|V(H)|^{3/2}$. In fact, Alon, Seymour, and Thomas [9] suspect that the correct factor for separators is $\Theta(|V(H)|)$, which holds e.g. in bounded-genus graphs. We also suspect that the same bound holds for the factor in Theorem 1.6, which would imply the corresponding bound for separators.

A third interesting direction is to generalize the polynomial-time approximation schemes that come out of bidimensionality to more general algorithmic problems that do not correspond directly to bidimensional parameters. One general family of such problems arises when adding weights to vertices and/or edges, and the goal is e.g. to find the minimum-weight dominating set. It is difficult to define bidimensionality of the corresponding weighted parameter because its value is no longer well-defined on an $r \times r$ grid: the parameter value now depends on the weights of vertices in such a grid. Another family of such problems arises when placing constraints (e.g., on coverage or domination) only on subsets of vertices and/or edges. Examples of such problems include Steiner tree [17] and subset feedback vertex set [89]. Again it is difficult to define bidimensionality in such cases because the value of the parameter on a grid depends on which vertices and/or edges of the grid are in the subset.

Bibliography

[1] P. K. Agarwal and C. M. Procopiuc, *Exact and approximation algorithms for clustering*, Algorithmica, 33 (2002), pp. 201–226.

[2] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.

[3] J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. A. Rosamond, and U. Stege, *Refined search tree technique for DOMINATING SET on planar graphs*, in Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science, vol. 2136 of Lecture Notes in Computer Science, 2001, pp. 111–122.

[4] J. Alber, M. R. Fellows, and R. Niedermeier, *Efficient data reduction for dominating set: A linear problem kernel for the planar case*, in The 8th Scandinavian Workshop on Algorithm Theory—SWAT 2002 (Turku, Finland), Springer, vol. 2368, Berlin, 2002, pp. 150–159.

[5] J. Alber, H. Fernau, and R. Niedermeier, *Graph separators: a parameterized view*, Journal of Computer and System Sciences, 67 (2003), pp. 808–832.

[6] J. Alber, H. Fernau, and R. Niedermeier, *Parameterized complexity: exponential speed-up for planar graph problems*, Journal of Algorithms, 52 (2004), pp. 26–56.

[7] J. ALBER AND J. FIALA, *Geometric separation and exact solutions for the parameterized independent set problem on disk graphs*, in Foundations of Information Technology in the Era of Networking and Mobile Computing, IFIP 17th WCC/TCS'02, Montréal, Canada, vol. 223 of IFIP Conference Proceedings, Kluwer, 2002, pp. 26–37.

[8] J. ALBER AND R. NIEDERMEIER, *Improved tree decomposition based algorithms for domination-like problems*, in Latin American Theoretical Informatics—LATIN 2002 (Cancun, Mexico), Springer, Lecture Notes in Computer Science, vol. 2286, Berlin, 2002, pp. 613–627.

[9] N. ALON, P. SEYMOUR, AND R. THOMAS, *A separator theorem for for graphs with excluded minor and its applications*, in Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (Baltimore, MD, 1990), 1990, pp. 293–299.

[10] ——, *A separator theorem for nonplanar graphs*, J. Amer. Math. Soc., 3 (1990), pp. 801–808.

[11] E. AMIR, *Efficient approximation for triangulation of minimum treewidth*, in Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI-2001), San Francisco, CA, 2001, Morgan Kaufmann Publishers, pp. 7–15.

[12] E. AMIR, R. KRAUTHGAMER, AND S. RAO, *Constant factor approximation of vertex-cuts in planar graphs*, in Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, ACM Press, 2003, pp. 90–99.

[13] T. ANDREAE, M. SCHUGHART, AND Z. TUZA, *Clique-transversal sets of line graphs and complements of line graphs*, Discrete Mathematics, 88 (1991), pp. 11–20.

[14] S. ARNBORG, D. G. CORNEIL, AND A. PROSKUROWSKI, *Complexity of finding embeddings in a k-tree*, SIAM Journal on Algebraic and Discrete Methods, 8 (1987), pp. 277–284.

[15] S. Arnborg, J. Lagergren, and D. Seese, *Problems easy for tree-decomposable graphs (extended abstract)*, in Proceedings of the 15th International Colloquium of Automata, Languages and Programming (Tampere, 1988), vol. 317 of Lecture Notes in Comput. Sci, Springer, Berlin, 1988, pp. 38–51.

[16] S. Arnborg and A. Proskurowski, *Characterization and recognition of partial 3-trees*, SIAM Journal on Algebraic and Discrete Methods, 7 (1986), pp. 305–314.

[17] S. Arora, M. Grigni, D. Karger, P. Klein, and A. Woloszyn, *A polynomial-time approximation scheme for weighted planar graph TSP*, in Proceedings of the 9th annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 1998, pp. 33–41.

[18] S. Arora, J. R. Lee, and A. Naor, *Euclidean distortion and the Sparsest Cut.* Manuscript, 2004.

[19] S. Arora, S. Rao, and U. Vazirani, *Expander flows, geometric embeddings, and graph partitionings*, in 36th Annual Symposium on the Theory of Computing, 2004.

[20] T. Asano, *An approach to the subgraph homeomorphism problem*, Theoretical Computer Science, 38 (1985), pp. 249–267.

[21] Y. Aumann and Y. Rabani, *An O(log k) approximate min-cut max-flow theorem and approximation algorithm*, SIAM J. Comput., 27 (1998), pp. 291–301.

[22] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and approximation*, Springer-Verlag, Berlin, 1999.

[23] B. S. Baker, *Approximation algorithms for NP-complete problems on planar graphs*, Journal of the Association for Computing Machinery, 41 (1994), pp. 153–180.

[24] V. Balachandran, P. Nagavamsi, and C. P. Rangan, *Clique transversal and clique independence on comparability graphs*, Information Processing Letters, 58 (1996), pp. 181–184.

[25] D. W. Bange, A. E. Barkauskas, L. H. Host, and P. J. Slater, *Generalized domination and efficient domination in graphs*, Discrete Mathematics, 159 (1996), pp. 1–11.

[26] J. Bar-Ilan, G. Kortsarz, and D. Peleg, *How to allocate network centers*, J. Algorithms, 15 (1993), pp. 385–415.

[27] A. Becker, R. Bar-Yehuda, and D. Geiger, *Randomized algorithms for the loop cutset problem*, Journal of Artificial Intelligence Research, 12 (2000), pp. 219–234.

[28] S. N. Bhatt and F. T. Leighton, *A framework for solving VLSI graph layout problems*, J. Comput. System Sci., 28 (1984), pp. 300–343.

[29] H. L. Bodlaender, *Dynamic programming algorithms on graphs with bounded treewidth*, in Proceedings of the 15th International Colloquium on Automata, Languages and Programming (Tampere, 1988), vol. 317 of Lecture Notes in Comput. Sci, Springer, Berlin, 1988, pp. 105–119.

[30] ——, *On disjoint cycles*, in Proceedings of the 18th International Workshop on Graph-Theoretic Concepts in Computer Science (Fischbachau, 1991), vol. 657 of Lecture Notes in Computer Science, Springer, Berlin, 1992, pp. 230–238.

[31] H. L. Bodlaender, *A tourist guide through treewidth*, Acta Cybernetica, 11 (1993), pp. 1–23.

[32] H. L. Bodlaender, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM Journal on Computing, 25 (1996), pp. 1305–1317.

[33] ——, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, 209 (1998), pp. 1–45.

[34] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, *Approximating treewidth, pathwidth, frontsize, and shortest elimination tree*, Journal of Algorithms, 18 (1995), pp. 238–255.

[35] H. L. Bodlaender and T. Kloks, *Efficient and constructive algorithms for the pathwidth and treewidth of graphs*, Journal of Algorithms, 21 (1996), pp. 358–402.

[36] H. L. Bodlaender, T. Kloks, and D. Kratsch, *Treewidth and pathwidth of permutation graphs*, SIAM Journal on Discrete Mathematics, 8 (1995), pp. 606–616.

[37] H. L. Bodlaender and R. H. Möhring, *The pathwidth and treewidth of cographs*, SIAM Journal on Discrete Mathematics, 6 (1993), pp. 181–188.

[38] H. L. Bodlaender and D. M. Thilikos, *Treewidth for graphs with small chordality*, Discrete Applied Mathematics, 79 (1997), pp. 45–61.

[39] T. Böhme, J. Maharry, and B. Mohar, *$K_{a,k}$ minors in graphs of bounded tree-width*, Journal of Combinatorial Theory, Series B, 86 (2002), pp. 133–147.

[40] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, American Elsevier Publishing Co., Inc., New York, 1976.

[41] V. Bouchitté, D. Kratsch, H. Mller, and I. Todinca, *On treewidth approximations*, in Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW'01).

[42] V. Bouchitté and I. Todinca, *Treewidth and minimum fill-in: grouping the minimal separators*, SIAM Journal on Computing, 31 (2001), pp. 212–232.

[43] J. Bourgain, *On Lipschitz embedding of finite metric spaces in Hilbert space*, Israel J. Math., 52 (1985), pp. 46–52.

[44] H. J. Broersma, E. Dahlhaus, and T. Kloks, *A linear time algorithm for minimum fill-in and treewidth for distance hereditary graphs*, Discrete Applied Mathematics, 99 (2000), pp. 367–400.

[45] T. N. Bui and C. Jones, *Finding good approximate vertex and edge partitions is np-hard*, Inf. Process. Lett., 42 (1992), pp. 153–159.

[46] T. N. Bui and A. Peck, *Partitioning planar graphs*, SIAM Journal on Computing, 21 (1992), pp. 203–215.

[47] J. F. Buss and J. Goldsmith, *Nondeterminism within P*, SIAM Journal on Computing, 22 (1993), pp. 560–572.

[48] L. Cai, M. Fellows, D. Juedes, and F. Rosamond, *On efficient polynomial-time approximation schemes for problems on planar structures.* Manuscript, 2001.

[49] M.-S. Chang, Y.-H. Chen, G. J. Chang, and J.-H. Yan, *Algorithmic aspects of the generalized clique-transversal problem on chordal graphs*, Discrete Applied Mathematics, 66 (1996), pp. 189–203.

[50] M.-S. Chang, T. Kloks, and C.-M. Lee, *Maximum clique transversals*, in Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2001, Boltenhagen, Germany), vol. 2204 of Lecture Notes in Computer Science, Berlin, 2001, Springer, pp. 32–43.

[51] M. Charikar and A. Sahai, *Dimension reduction in the l_1 norm*, in Proceedings of the 43th Annual Symposium on Foundations of Computer Science (FOCS'02), 2002, pp. 551–560.

[52] S. Chawla, A. Gupta, and H. Raecke, *Embeddings of negative-type metrics and improved approximations to sparsest cut*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, Canada, January 2005. To appear.

[53] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and S. Alistair, *Embedding k-outerplanar graphs into ℓ_1*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03), 2003, pp. 527–536.

[54] C. Chekuri, S. Khanna, and B. Shepherd, *Edge-disjoint paths in planar graphs*, in Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS'04), 2004, pp. 71–80.

[55] J. Chen, I. A. Kanj, and W. Jia, *Vertex cover: further observations and further improvements*, Journal of Algorithms, 41 (2001), pp. 280–301.

[56] Z.-Z. Chen, *Efficient approximation schemes for maximization problems on $K_{3,3}$-free or K_5-free graphs*, Journal of Algorithms, 26 (1998), pp. 166–187.

[57] ——, *Approximation algorithms for independent sets in map graphs*, J. Algorithms, 41 (2001), pp. 20–40.

[58] Z.-Z. Chen, M. Grigni, and C. H. Papadimitriou, *Map graphs*, Journal of the ACM, 49 (2002), pp. 127–138.

[59] N. Chiba, T. Nishizeki, and N. Saito, *An approximation algorithm for the maximum independent set problem on planar graphs*, SIAM Journal on Computing, 11 (1982), pp. 663–675.

[60] E. J. Cockayne, S. Goodman, and S. Hedetniemi, *A linear time algorithm for the domination number of a tree*, Information Processing Letters, 4 (1975), pp. 41–44.

[61] B. Courcelle, *Graph rewriting: an algebraic and logic approach*, in Handbook of Theoretical Computer Science, Vol. B, Elsevier, Amsterdam, 1990, pp. 193–242.

[62] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Bidimensional parameters and local treewidth*, SIAM Journal on Discrete Mathematics, 18 (2004), pp. 501–511.

[63] ——, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA'04), January 2004, pp. 823–832.

[64] ——, *Fixed-parameter algorithms for the (k, r)-center in planar graphs and map graphs*, ACM Transactions on Algorithms, (2005). To appear. A preliminary version appears in *Proceedings of the 30th International Colloquium on Automata, Languages and Programming*, LNCS 2719, 2003, pages 829–844.

[65] E. D. Demaine and M. Hajiaghayi, *Diameter and treewidth in minor-closed graph families, revisited*, Algorithmica, 40 (2004), pp. 211–215.

[66] E. D. Demaine and M. Hajiaghayi, *Equivalence of local treewidth and linear local treewidth and its algorithmic applications*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA'04), January 2004, pp. 833–842.

[67] E. D. Demaine and M. Hajiaghayi, *Fast algorithms for hard graph problems: Bidimensionality, minors, and local treewidth*, in Proceedings of the 12th International Symposium on Graph Drawing, vol. 3383 of Lecture Notes in Computer Science, Harlem, NY, 2004, pp. 517–533.

[68] ——, *Quickly deciding minor-closed parameters in general graphs*. Manuscript, 2004.

[69] E. D. Demaine and M. Hajiaghayi, *Bidimensionality, map graphs, and grid minors*. Submitted, February 2005.

[70] E. D. Demaine and M. Hajiaghayi, *Bidimensionality: New connections between FPT algorithms and PTASs*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, January 2005, pp. 590–601.

[71] ——, *Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality*, in Pro-

ceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, January 2005, pp. 682–689.

[72] E. D. Demaine, M. Hajiaghayi, N. Nishimura, P. Ragde, and D. M. Thilikos, *Approximation algorithms for classes of graphs excluding single-crossing graphs as minors*, Journal of Computer and System Sciences, 69 (2004), pp. 166–195.

[73] E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos, *The bidimensional theory of bounded-genus graphs*, in Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS 2004), Prague, August 2004, pp. 191–203.

[74] E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos, *Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors*, Algorithmica, 41 (2005), pp. 245–267.

[75] M. DeVos, G. Ding, B. Oporowski, D. P. Sanders, B. Reed, P. Seymour, and D. Vertigan, *Excluding any graph as a minor allows a low tree-width 2-coloring*, Journal of Combinatorial Theory. Series B, 91 (2004), pp. 25–41.

[76] J. Díaz, M. J. Serna, and J. Torán, *Parallel approximation schemes for problems on planar graphs*, Acta Informatica, 33 (1996), pp. 387–408.

[77] R. Diestel, *Simplicial decompositions of graphs: a survey of applications*, Discrete Mathematics, 75 (1989), pp. 121–144. Graph theory and combinatorics (Cambridge, 1988).

[78] ——, *Decomposing infinite graphs*, Discrete Mathematics, 95 (1991), pp. 69–89. Directions in infinite graph theory and combinatorics (Cambridge, 1989).

[79] ——, *Graph theory*, vol. 173 of Graduate Texts in Mathematics, Springer-Verlag, New York, second ed., 2000.

[80] R. Diestel, T. R. Jensen, K. Y. Gorbunov, and C. Thomassen, *Highly connected sets and the excluded grid theorem*, Journal of Combinatorial Theory, Series B, 75 (1999), pp. 61–73.

[81] R. Diestel and R. Thomas, *Excluding a countable clique*, Journal of Combinatorial Theory. Series B, 76 (1999), pp. 41–67.

[82] I. Dinur and S. Safra, *The importance of being biased*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing (Montréal, 2002), 2002, pp. 33–42.

[83] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer-Verlag, New York, 1999.

[84] J. Ellis, H. Fan, and M. Fellows, *The dominating set problem is fixed parameter tractable for graphs of bounded genus*, Journal of Algorithms, 52 (2004), pp. 152–168.

[85] D. Eppstein, *Subgraph isomorphism in planar graphs and related problems*, in Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1995), New York, 1995, ACM, pp. 632–640.

[86] ——, *Subgraph isomorphism in planar graphs and related problems*, Journal of Graph Algorithms and Applications, 3 (1999), pp. no. 3, 27 pp.

[87] ——, *Diameter and treewidth in minor-closed graph families*, Algorithmica, 27 (2000), pp. 275–291.

[88] G. Even, J. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, 20 (1998), pp. 151–174.

[89] G. Even, J. Naor, and L. Zosin, *An 8-approximation algorithm for the subset feedback vertex set problem*, SIAM Journal on Computing, 30 (2000), pp. 1231–1252.

[90] U. Feige, M. Hajiaghayi, and J. R. Lee, *Improved approximation algorithms for minimum-weight vertex separators*, in Proceedings of the 37th ACM Symposium on Theory of Computing (STOC 2005), Baltimore, May 2005. To appear.

[91] M. R. Fellows and M. A. Langston, *Nonconstructive tools for proving polynomial-time decidability*, Journal of the ACM, 35 (1988), pp. 727–739.

[92] F. V. Fomin and D. M. Thilikos, *New upper bounds on the decomposability of planar graphs and fixed parameter algorithms*, tech. report, Universitat Politècnica de Catalunya, Spain, 2002.

[93] ——, *Dominating sets in planar graphs: Branch-width and exponential speed-up*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 168–177.

[94] F. V. Fomin and D. M. Thilikos, *Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up*, in Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004), Turku, Finland, July 2004, pp. 581–592.

[95] A. S. Fraenkel, *Planar kernel and Grundy with $d \leq 3$, $d_{\text{out}} \leq 2$, $d_{\text{in}} \leq 2$ are NP-complete*, Discrete Applied Mathematics, 3 (1981), pp. 257–262.

[96] M. Frick and M. Grohe, *Deciding first-order properties of locally tree-decomposable structures*, Journal of the ACM, 48 (2001), pp. 1184–1206.

[97] H. Friedman, N. Robertson, and P. Seymour, *The metamathematics of the graph minor theorem*, in Logic and combinatorics (Arcata, Calif., 1985), vol. 65 of Contemp. Math., Amer. Math. Soc., Providence, RI, 1987, pp. 229–261.

[98] H. Gaifman, *On local and nonlocal properties*, in Proceedings of the Herbrand symposium (Marseilles, 1981), vol. 107 of Stud. Logic Found. Math., Amsterdam, 1982, North-Holland, pp. 105–135.

[99] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Co., San Francisco, Calif., 1979.

[100] C. GAVOILLE, D. PELEG, A. RASPAUD, AND E. SOPENA, *Small k-dominating sets in planar graphs with applications*, in Graph-theoretic concepts in computer science (Boltenhagen, 2001), vol. 2204 of Lecture Notes in Comput. Sci., Springer, Berlin, 2001, pp. 201–216.

[101] J. R. GILBERT, J. P. HUTCHINSON, AND R. E. TARJAN, *A separator theorem for graphs of bounded genus*, J. Algorithms, 5 (1984), pp. 391–407.

[102] T. F. GONZALEZ, *Clustering to minimize the maximum intercluster distance*, Theoret. Comput. Sci., 38 (1985), pp. 293–306.

[103] M. GROHE, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), pp. 613–632.

[104] A. GUPTA, I. NEWMAN, Y. RABINOVICH, AND S. ALISTAIR, *Cuts, trees and ℓ_1-embeddings of graphs*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science, 1999, pp. 399–409.

[105] A. GUPTA AND N. NISHIMURA, *Sequential and parallel algorithms for embedding problems on classes of partial k-trees*, in Algorithm theory—Scandinavian Workshop on Algorithm Theory 1994 (Aarhus, 1994), Springer, Berlin, 1994, pp. 172–182.

[106] V. GURUSWAMI AND C. P. RANGAN, *Algorithmic aspects of clique-transversal and clique-independent sets*, Discrete Applied Mathematics, 100 (2000), pp. 183–202.

[107] G. GUTIN, T. KLOKS, AND C. M. LEE, *Kernels in planar digraphs*, in Optimization Online, Mathematical Programming Society, Philadelphia, 2001.

[108] M. Habib and R. H. Möhring, *Treewidth of cocomparability graphs and a new order-theoretic parameter*, ORDER, 1 (1994), pp. 47–60.

[109] M. Hajiaghayi, *Algorithms for Graphs of (Locally) Bounded Treewidth*, master's thesis, University of Waterloo, September 2001.

[110] M. Hajiaghayi and N. Nishimura, *Subgraph isomorphism, log-bounded fragmentation and graphs of (locally) bounded treewidth*, in Proc. the 27th International Symposium on Mathematical Foundations of Computer Science (Poland, 2002), 2002, pp. 305–318.

[111] L. H. Harper, *Optimal numberings and isoperimetric problems on graphs*, J. Combinatorial Theory, 1 (1966), pp. 385–393.

[112] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker Inc., 1998.

[113] I. V. Hicks, *Branch Decompositions and their applications*, PhD thesis, Rice University, 2000.

[114] J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. Comput., 2 (1973), pp. 135–158.

[115] A. Kanevsky and V. Ramachandran, *Improved algorithms for graph four-connectivity*, Journal of Computer and System Sciences, 42 (1991), pp. 288–306. Twenty-Eighth IEEE Symposium on Foundations of Computer Science (Los Angeles, CA, 1987).

[116] I. Kanj and L. Perković, *Improved parameterized algorithms for planar dominating set*, in Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science, vol. 2420 of Lecture Notes in Computer Science, Springer, Lecture Notes in Computer Science, Berlin, vol.2420, 2002, pp. 399–410.

[117] J. A. KELNER, *Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus*, in Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, ACM Press, 2004, pp. 455–464.

[118] A. KÉZDY AND P. MCGUINNESS, *Sequential and parallel algorithms to find a K_5 minor*, in Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, 1992), 1992, pp. 345–356.

[119] S. KHOT AND N. VISHNOI, *On embeddability of negative type metrics into ℓ_1*. Manuscript, 2004.

[120] P. N. KLEIN, S. A. PLOTKIN, AND S. RAO, *Excluded minors, network decomposition, and multicommodity flow*, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing, 1993, pp. 682–690.

[121] T. KLOKS, *Treewidth of circle graphs*, in Algorithms and computation (Hong Kong, 1993), Springer, Berlin, 1993, pp. 108–117.

[122] T. KLOKS AND L. CAI, *Parameterized tractability of some (efficient) Y-domination variants for planar graphs and t-degenerate graphs*, in Proceedings of the International Computer Symposium (ICS 2000), Taiwan, 2000.

[123] T. KLOKS, C. M. LEE, AND J. LIU, *New algorithms for k-face cover, k-feedback vertex set, and k-disjoint set on plane and planar graphs*, in Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2002), vol. 2573 of Lecture Notes in Computer Science, 2002, pp. 282–295.

[124] R. KRAUTHGAMER, J. R. LEE, M. MENDEL, AND A. NAOR, *Measured descent: A new embedding method for finite metrics*, in 45th Symposium on Foundations of Computer Science, 2004. To appear.

[125] J. LAGERGREN, *Efficient parallel algorithms for graphs of bounded tree-width*, Journal of Algorithms, 20 (1996), pp. 20–44.

[126] J. R. LEE, *On distance scales, embeddings, and efficient relaxations of the cut cone*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, Vancouver, 2005, ACM. To appear.

[127] J. V. LEEUWEN, *Graph algorithms*, in Handbook of theoretical computer science, Vol. A, Elsevier, Amsterdam, 1990, pp. 525–631.

[128] F. T. LEIGHTON, *Complexity Issues in VLSI: Optimal Layout for the Shuffle-Exchange Graph and Other Networks*, MIT Press, Cambridge, MA, 1983.

[129] T. LEIGHTON AND S. RAO, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, J. ACM, 46 (1999), pp. 787–832.

[130] C. LEISERSON, *Area-efficient graph layouts (for VLSI)*, in 21th Annual Symposium on Foundations of Computer Science, IEEE Computer Soc., Los Alamitos, CA, 1980, pp. 270–280.

[131] N. LINIAL, E. LONDON, AND Y. RABINOVICH, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15 (1995), pp. 215–245.

[132] R. J. LIPTON AND R. E. TARJAN, *Applications of a planar separator theorem*, SIAM Journal on Computing, 9 (1980), pp. 615–627.

[133] J. MATOUŠEK AND R. THOMAS, *On the complexity of finding iso- and other morphisms for partial k-trees*, Discrete Mathematics, 108 (1992), pp. 343–364. Topological, algebraical and combinatorial structures. Frolík's memorial volume.

[134] G. L. MILLER AND V. RAMACHANDRAN, *A new graph triconnectivity algorithm and its parallelization*, Combinatorica, 12 (1992), pp. 53–76.

[135] G. L. MILLER, S.-H. TENG, W. THURSTON, AND S. A. VAVASIS, *Separators for sphere-packings and nearest neighbor graphs*, J. ACM, 44 (1997), pp. 1–29.

[136] B. Mohar and C. Thomassen, *Graphs on surfaces*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2001.

[137] J. Plesník, *On the computational complexity of centers locating in a graph*, Apl. Mat., 25 (1980), pp. 445–452. With a loose Russian summary.

[138] S. A. Plotkin, S. Rao, and W. D. Smith, *Shallow excluded minors and improved graph decompositions*, in Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'94), 1994, pp. 462–470.

[139] Y. Rabinovich, *On average distortion of embedding metrics into the line and into L_1*, in 35th Annual ACM Symposium on Theory of Computing, ACM, 2003.

[140] S. Rao, *Small distortion and volume preserving embeddings for planar and Euclidean metrics*, in Proceedings of the 15th Annual Symposium on Computational Geometry, New York, 1999, ACM, pp. 300–306.

[141] B. A. Reed, *Tree width and tangles: a new connectivity measure and some applications*, in Surveys in combinatorics, 1997 (London), vol. 241 of London Math. Soc. Lecture Note Ser., Cambridge Univ. Press, Cambridge, 1997, pp. 87–162.

[142] N. Robertson and P. Seymour, *Excluding a graph with one crossing*, in Graph structure theory (Seattle, 1991), Amer. Math. Soc., Providence, RI, 1993, pp. 669–675.

[143] N. Robertson and P. D. Seymour, *Graph minors. II. Algorithmic aspects of tree-width*, Journal of Algorithms, 7 (1986), pp. 309–322.

[144] N. Robertson and P. D. Seymour, *Graph minors. V. Excluding a planar graph*, Journal of Combinatorial Theory, Series B, 41 (1986), pp. 92–114.

[145] ——, *Graph minors. X. Obstructions to tree-decomposition*, Journal of Combinatorial Theory, Series B, 52 (1991), pp. 153–190.

[146] ——, *Graph minors. XI. Circuits on a surface*, Journal of Combinatorial Theory, Series B, 60 (1994), pp. 72–106.

[147] ——, *Graph minors. XII. Distance on a surface*, Journal of Combinatorial Theory, Series B, 64 (1995), pp. 240–272.

[148] ——, *Graph minors. XIII. The disjoint paths problem*, Journal of Combinatorial Theory, Series B, 63 (1995), pp. 65–110.

[149] ——, *Graph minors. XVI. Excluding a non-planar graph*, Journal of Combinatorial Theory. Series B, 89 (2003), pp. 43–76.

[150] ——, *Graph minors. XX. Wagner's conjecture*, J. Combin. Theory Ser. B, 92 (2004), pp. 325–357.

[151] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, Journal of Combinatorial Theory, Series B, 62 (1994), pp. 323–348.

[152] D. J. Rose, *On simple characterizations of k-trees*, Discrete Math., 7 (1974), pp. 317–322.

[153] D. P. Sanders, *On linear recognition of tree-width at most four*, SIAM Journal on Discrete Mathematics, 9 (1996), pp. 101–117.

[154] P. D. Seymour, *Personal communication*, May 2004.

[155] P. D. Seymour and R. Thomas, *Call routing and the ratcatcher*, Combinatorica, 14 (1994), pp. 217–241.

[156] R. Sundaram, K. S. Singh, and P. C. Rangan, *Treewidth of circular-arc graphs*, SIAM Journal on Discrete Mathematics, 7 (1994), pp. 647–655.

[157] C. Swamy and A. Kumar, *The 5th international workshop on approximation algorithms for combinatorial optimization*, in The 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (Italy, APPROX 2002), LNCS, 2002, pp. 256–270.

[158] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

[159] J. A. Telle and A. Proskurowski, *Practical algorithms on partial k-trees with an application to domination-like problems*, in Proceedings of 3rd Workshop on Algorithms and Data Structures (Montréal, 1993), vol. 709 of Lecture Notes in Computer Science, Springer, Berlin, 1993, pp. 610–621.

[160] J. A. Telle and A. Proskurowski, *Algorithms for vertex partitioning problems on partial k-trees*, SIAM Journal on Discrete Mathematics, 10 (1997), pp. 529–550.

[161] A. Thomason, *The extremal function for complete minors*, Journal of Combinatorial Theory, Series B, 81 (2001), pp. 318–338.

[162] M. Thorup, *Map graphs in polynomial time*, in Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 396–407.

[163] R. Uehara, *NP-complete problems on a 3-connected cubic planar graph and their applications*, Tech. Report TWCU-M-0004, Tokyo Woman's Christian University, 1996.

[164] K. Wagner, *Über eine Eigenschaft der ebenen Komplexe*, Math. Ann., 114 (1937), pp. 570–590.

[165] D. B. West, *Introduction to Graph Theory*, Prentice Hall Inc., Upper Saddle River, NJ, 1996.

[166] S. G. Williamson, *Depth-first search and Kuratowski subgraphs*, Journal of the Association for Computing Machinery, 31 (1984), pp. 681–693.

[167] M. YANNAKAKIS, *Node- and edge-deletion NP-complete problems*, in Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, CA, 1978), ACM press, New York, 1978, pp. 253–264.