Graph Minors and Parameterized Algorithm Design[⋆]

Dimitrios M. Thilikos

Department of Mathematics, National and Kapodistrian University of Athens,
Panepistimioupolis, GR-15784, Athens, Greece
sedthilk@math.uoa.gr

Abstract. The Graph Minors Theory, developed by Robertson and Seymour, has been one of the most influential mathematical theories in parameterized algorithm design. We present some of the basic algorithmic techniques and methods that emerged from this theory. We discuss its direct meta-algorithmic consequences, we present the algorithmic applications of core theorems such as the grid-exclusion theorem, and we give a brief description of the irrelevant vertex technique.

Keywords: graph minors, parameterized algorithms, treewidth, bidimensionality, irrelevant vertex technique, linkages.

1 Introduction

Graph Minors Theory (GMT) was developed by Robertson and Seymour in a series of 23 papers, between 1984 and 2009. Among them, the second paper of the series was published in the Journal of Algorithms while all the rest were published in the Journal of Combinatorial Theory Series B. The main theoretical achievement of this project was the proof of Wagner's conjecture, now known as the *Robertson & Seymour Theorem*, stating that graphs are well-quasi-ordered under the minor containment relation. Besides its purely mathematical importance, GMT induced a series of powerful algorithmic results and techniques that had a deep influence on theoretical computer science. More particularly, GMT has been one of the most powerful "mathematical engines" in the theory and design of parameterized algorithms. In particular, a considerable part of the basic techniques in parameterized algorithm design is directly or indirectly linked to results from GMT. Moreover, GMT offered the theoretical base for the understanding and resolution of some of the most prominent graph-algorithmic problems in parameterized complexity. In what follows, we give a brief presentation of the main results and techniques in this area.

[⋆] This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: "*Thalis. Investing in knowledge society through the European Social Fund*".

H.L. Bodlaender et al. (Eds.): Fellows Festschrift, LNCS 7370, pp. 228–256, 2012.

Our presentation is organized as follows. In Section 2 we give the definitions of some basic combinatorial and algorithmic concepts. In Section 3.1 we present the main algorithmic consequences of the GMT, mainly from the parameterized complexity viewpoint. Section 4 is devoted to the celebrated grid-exclusion theorem and its applications to bidimensionality theory. Finally, Section 5, attempts a short presentation of the irrelevant vertex technique and its applications.

2 Basic Definitions

All graphs we consider are finite, undirected and simple, i.e., they do not have multiple edges or loops. Given a graph G we denote by $V(G)$ and $E(G)$ its vertex and edge set respectively. The *size* (reps. *magnetite*) of a graph G is the number of its vertices (reps. edges) and is denoted by $n(G)$ (reps. $m(G)$), i.e., $n(G) = |V(G)|$ ($m(G) = |E(G)|$). We denote by $G \setminus v$ the graph obtained by removing v (along with its incident edges) from G. The *neighborhood* of a vertex $v \in V(G)$, denoted by $N_G(v)$, is the set of edges in G that are adjacent to v. The *degree* of a vertex $v \in V(G)$ is the cardinality of its neighborhood in G. We denote by K_r the complete graph on r vertices and by $K_{r,q}$ the complete bipartite graph with r vertices in its one part and q in the other. Finally, we denote by G_k the $(k \times k)$-grid, i.e., the Cartesian product of two paths of length $k-1$ (see Figure 1).

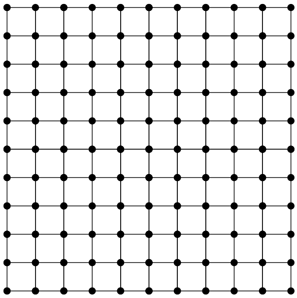

Fig. 1. The $(11, 11)$-grid G_{11}

2.1 Relations on Graphs and Obstructions

We say that a graph H is a *subgraph* of a graph G if H can be obtained by G by removing edges or vertices. The contraction of an edge $e = \{x, y\}$ from G is the removal from G of all edges incident to x or y and the insertion of a new vertex v_e that is made adjacent to all the vertices of $(N_G(x) \setminus \{y\}) \cup (N_G(y) \setminus \{x\})$. Given two graphs H and G, we say that H is a *contraction* of G, denoted by $H \leq_c G$, if H can be obtained from G by a (possibly empty) series of edge contractions.

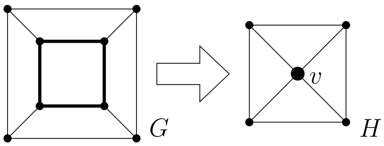

Fig. 2. The graph H is the result of the contractions of the bold edges in G to the vertex v

H is a *minor* of G if H is a contraction of some subgraph of G. A graph H is a *topological minor* of G (denoted by $H \leq_t G$) if G contains as a subgraph some subdivision of H (a *subdivision* of a graph H is any graph obtained by replacing some of its edges by paths between the same endpoints). Given a partial ordering relation $\leq$ on graphs, we say that a graph class $\mathcal{G}$ is *closed under* $\leq$ if for every $G \in \mathcal{G}$, $H \leq G$ implies that $H \in \mathcal{G}$. Let $\mathcal{G}$ be a graph class that is closed under the minor relation. An $\leq$-*anti-chain* is a set of graphs that are pairwise non-comparable with respect to $\leq$. For example the set of graphs $\mathcal{A} = \{K_{2,r} \mid r \geq 2\}$ is a $\leq_c$-antichain but not an $\leq_m$-antichain or a $\leq_t$-antichain.

We define the $\leq$-*obstruction set* of a graph class $\mathcal{G}$, denoted by $\mathbf{obs}_{\leq}(\mathcal{G})$, as the set of all $\leq$-minimal graphs that *do not* belong to $\mathcal{G}$. Clearly, by definition, the $\leq$-obstruction set of a graph class is an $\leq$-anti-chain. Obstruction sets can be seen as alternative characterizations of graph classes and, in many cases, reveal a good deal from their structural characteristics. For example, it easy to verify that $\mathbf{obs}_{\leq_m}(\mathcal{T}) = \mathbf{obs}_{\leq_t}(\mathcal{T}) = \{K_3\}$, $\mathbf{obs}_{\leq_m}(\mathcal{O}) = \mathbf{obs}_{\leq_t}(\mathcal{O}) = \{K_4, K_{2,3}\}$, where $\mathcal{T}$ and $\mathcal{O}$ are the classes of all acyclic and all outerplanar graphs respectively, and $\mathbf{obs}_{\leq_c}(\mathcal{O}^*) = \{K_4, K_{2,3}, K_{2,3}^+\}$ where $\mathcal{O}^*$ are the connected outerplanar graphs and $K_{2,3}^+$ is the graph obtained from K_5 by removing a triangle. The most classic theorems on obstruction characterization of graph classes are the Kuratowski-Pontryagin's theorem [87] and Wagner's theorem [121], stating that $\mathbf{obs}_{\leq_m}(\mathcal{P}) = \{K_5, K_{3,3}\}$ and $\mathbf{obs}_{\leq_t}(\mathcal{P}) = \{K_5, K_{3,3}\}$ respectively, where $\mathcal{P}$ is the class of all planar graphs.

2.2 Parameterized Problems and Algorithms

The idea of problem parameterization is to treat algorithmic problems as parameterized entities and to evaluate the complexity of the corresponding algorithms by considering the way parameters appear in their running times. As here we deal with problems on graphs, we adapt the classic definitions of parameterized complexity to the case where problem inputs represent graphs.

Formally, a *parameterized problem on graphs* is a subset Π of $\Sigma^* \times \mathbb{N}$, where Σ is some alphabet and, in each $(I, k) \in \Sigma^* \times \mathbb{N}$, I encodes a combinatorial structure related to one, or more, graphs. For this, we agree that n (resp. m) is the maximum size (resp. magnitude) of the graphs encoded in I and we insist that $|(I, k)| = O(m)$. We call I the *main part of the input* and we say that k is the *parameter* of the problem. Two instances $(I, k), (I', k') \in \Sigma^* \times \mathbb{N}$ are *equivalent with respect to* Π if $(I, k) \in \Pi \iff (I', k') \in \Pi$.

We say that Π is *fixed parameter tractable* if there exists a function[1] $f : \mathbb{N} \to \mathbb{N}$ and an algorithm deciding whether $(I, k) \in \Pi$ in $O(f(k) \cdot n^c)$ steps, where c is a constant not depending on the parameter k of the problem. We call such an algorithm FPT-*algorithm* or, to express concretely the choice of f and c, we say that $\Pi \in O(f(k) \cdot n^c)$-$\mathsf{FPT}$. A parameterized problem on graphs belongs to the parameterized class FPT if it can be solved by an FPT-algorithm. In fact, not all parameterized problems belong to the class FPT. There is a hierarchy of parameterized complexity classes, namely

$$\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \mathsf{W}[3] \subseteq \ldots \subseteq \mathsf{W}[SAT] \subseteq \mathsf{W}[\mathrm{P}] \subseteq \mathsf{XP},$$

and appropriate *parameter-preserving* reductions such that, when a problem is hard for some of them (other than FPT), it is not expected to have an FPT-algorithm (all inclusions in this hierarchy are believed to be strict). See the monographs [39, 44, 97] for more details on parameterized complexity.

Time bounds for parameterized algorithms have two parts. The term $f(k)$ is called *parameter dependence* and, is typically a super-polynomial function. On the other hand, the term n^c is a polynomial function and we call it *polynomial part*. In most of the problems that we examine here, I will encode a simple graph. To simplify notation, we frequently write "$O_k(n^c)$" instead of "$f(k) \cdot (n^c)$" for some *recursive* function $f : \mathbb{N} \to \mathbb{N}$" and, in this case, we refer to the function f hidden in the O_k notation as the *parameter dependence.*

3 Algorithmic Consequences of the GMT

3.1 Well-Quasi-Ordering

The main combinatorial result of the GMT fits in the more general framework of the theory of Well-Quasi-Orderings, first developed by Graham Higman[2] under the name "finite basis property" [65]. Given a set $\mathcal{X}$ and a partial ordering $\leq$ on $\mathcal{X}$ we say that $\mathcal{X}$ is *well-quasi-ordered under* $\leq$ if none of its subsets is an infinite $\leq$-antichain.

Theorem 1 (Robertson & Seymour Theorem [110]). *The set of all graphs is well-quasi-ordered under minors.*

In other words, Theorem 1 says that If $\mathcal{G}$ is an infinite set of graphs then there exist two graphs $H, G \in \mathcal{G}$ such that H is a minor of G. The proof of theorem 1 was concluded in paper XX of the Graph Minors Series. Before its proof, the statement of Theorem 1 was known as Wagner's conjecture. However, as mentioned by Diestel in [32], Wagner said that he had never made such a conjecture. A similar conjecture, on the well-quasi-ordering of trees under the topological minor

[1] Notice that in the definition of FPT f is not necessarily a recursive function.

[2] As mentioned in [86], the same theory was also developed in some unpublished manuscript of Erdős and Rado, while its first hints can be traced back to B. H. Neumann [96]

relation was made by Vázsonyi and was proved in 1960 independently[3] by Joseph Kruskal and S. Tarkowski [91]. Interesting results on the meta-mathematics of Kruskal's tree theorem as well as Roberson & Seymour's theorem can be found in [55] and [54] respectively.

Consider the following parameterized problem:

H-Minor Checking
Instance: Two graphs G and H.
Parameter: $k = |V(H)|$.
Question: Is H a minor of G?

The main algorithmic contribution of the GMT is the following result.

Theorem 2 (Robertson and Seymour [108]). *One can construct an algorithm that, given a n-vertex graph G and a k-vertex graph H, checks whether H is a minor of a graph G in $O_k(n^3)$ steps. In other words, H-Minor Checking $\in O_k(n^3)$-FPT.*

Actually, Robertson and Seymour in [108] describe an $O_k(n^3)$-step algorithm that solves a generalization of the H-Minor Checking and another celebrated problem, namely the k-Disjoint Paths problem. In Section 5, we give a rough description of the main ingredients of the algorithm in Theorem 2 especially for the k-Disjoint Paths problem. Recently, this running time was improved to a quadratic one for the k-Disjoint Paths problem in [74].

The good news about Theorem 2 is that it is constructive (contrary to Theorem 1) and there is a recursive function hidden in the O_k notation. The bad news is that, according to the algorithm in [108] and the proof of its correctness in [111] and [107], the values of this function are *immense*[4], even for small values of k. David Johnson mentioned in [67]:

> "*for any instance $G = (V, E)$ that one could fit into the known universe, one would easily prefer $|V|^{70}$ to even constant time, if that constant had to be one of Robertson and Seymour's*".

Moreover, in [67], David Johnson estimates that just one constant in the parameter dependence of Theorem 2 is roughly

$$2^{\uparrow 2^{2^{2^{\uparrow 2^{\uparrow \Theta(r)}}}}}$$

where $2^{\uparrow r}$ denotes a tower $2^{2^{2^{\cdot^{\cdot^{\cdot}}}}}$ involving r 2's. Clearly, such type of constants may create reasonable doubts to computer scientists on whether such an algorithm may be considered to be an "algorithm" of some practical meaning. In fact, to investigate until which point these constants can be improved is an open and challenging problem in parameterized complexity and algorithms (see e.g. [3]).

[3] A shorter and quite elegant proof of Vázsonyi's conjecture was given by Nash-Williams in 1963 [95].

[4] Perhaps the word "immense" is somehow moderate here. Instead, Fellows and Langston used the expression "mind-boggling" in [43].

3.2 Minor-Closed Graph Parameters

A *parameter on graphs* (or *a graph parameter*) is any function that maps graphs to integers and with the property that it is invariant under graph isomorphism. Let $\leq$ be a relation on graphs. We say that a graph parameter $\mathbf{p}$ is *closed under* $\leq$ (or, simply, $\leq$*-closed*) if for every two graph H and G, $H \leq G$ implies that $\mathbf{p}(H) \leq \mathbf{p}(G)$. We define the $\leq$*-obstruction family* of $\mathbf{p}$ as the parameterized graph class

$$\mathcal{O}^{\leq}_{\mathbf{p},k} = \mathbf{obs}_{\leq}(\{G \mid \mathbf{p}(G) \leq k\}).$$

Consider the following parameterized meta-problem.

k-PARAMETER CHECKING FOR $\mathbf{p}$
Instance: a graph G and an integer $k \geq 0$.
Parameter: k
Question: $\mathbf{p}$(G)$\leq k$?

Theorems 1 and 2 together have the following dramatic consequence.

Theorem 3. *For every parameter* $\mathbf{p}$ *that is closed under minors there exists an algorithm that solves the problem* k-PARAMETER CHECKING FOR $\mathbf{p}$ *in* $f(k) \cdot n^3$ *steps for some function* f.

Proof. Recall that, by definition, no two graphs in $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ can be comparable graphs under the minor relation. It follows, from Theorem 1, that $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ is a finite set. Let $g(k) = |\mathcal{O}^{\leq_m}_{\mathbf{p},k}|$. As $\mathbf{p}$ is closed under minors, it holds that

$$\mathbf{p}(G) \leq k \iff \forall H \in \mathcal{O}^{\leq_m}_{\mathbf{p},k} \quad H \not\leq_m G.$$

Therefore, to check whether $\mathbf{p}(G) \leq k$ it is enough to apply $g(k)$ times the $O_k(n^3)$ step algorithm of Theorem 2 and check whether some member of $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ is contained as a minor in G.

Theorem 3 had a great impact in parameterized complexity as it implied a massive classification of problems in the class FPT. In that sense, Theorem 3 is an algorithmic meta-theorem because it provides a generic condition (minor-closedness) for a parameterized problem that automatically implies the existence of an FPT-algorithm for it. Unfortunately, the proof of Theorem 1 does not provide any general "meta-algorithm" to compute the set $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ and, that way, construct the claimed algorithm for each $\mathbf{p}$. In fact, due to the meta-mathematics of Theorem 1 [54], such an meta-algorithm does not exist. As observed in [43], there is no algorithm that, given a Turing machine accepting precisely the graphs of a minor-closed graph class $\mathcal{F}$, outputs $\mathbf{obs}_{\leq_m}(\mathcal{F})$ (see also [119]). However, Theorem 3 gave important (mathematical) energy to Parameterized Algorithms as it acted as an "encouraging factor". The knowledge that an algorithm exists for a specific problem, induces the challenge to construct one and, in a sense, provides the courage to try to accomplish such a task.

In order to cope with the inherent non-constructivity of Theorem 3 one may study specific parameters where the computation of the set $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ (or, at least, of some upper bound to the function $g(k)$) in the proof of Theorem 3 is possible. However, this is not an easy task, even for simple parameters. According to [34], if the problem of checking whether $\mathbf{p}(G) \leq k$ is NP-complete, then $|\mathcal{O}^{\leq_m}_{\mathbf{p},k}|$ is a super-polynomial function of k, unless the polynomial hierarchy collapses to Σ_3^{P}. Characterizations of $\mathbf{p}(G) \leq k$ (yielding better lower bounds for $|\mathcal{O}^{\leq_m}_{\mathbf{p},k}|$) have been provided for several parameters [10, 20, 40, 58, 84, 98, 99, 115, 117, 118]. However, to our knowledge, there is not yet a natural parameter $\mathbf{p}$ for which a complete characterization of $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ is known. A more promising strategy towards detecting constructive fragments of Theorem 3, is to detect parameters – or families of parameters – where $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$ is recursive. For this one may either prove upper bounds for $|\mathcal{O}^{\leq_m}_{\mathbf{p},k}|$, as done in [57] for the case of branchwidth[5], or provide partial characterizations of $\mathcal{O}^{\leq_m}_{\mathbf{p},k}$, as done in [2, 19, 21, 89, 94], that permit its recursive computation.

At this point, we should mention that all theorems of this section have their counterparts in another partial relation on graphs, the one of *immersion*. The *lift* of two incident edges is the operation of removing two edges $e_1 = \{x, y\}$ and $e_2 = \{x, z\}$ (incident to a common vertex x) and adding the edge $\{y, z\}$. We say that a graph H can be *immersed* in a graph G, denoted by $H \leq_{im} G$, if H can be obtained from a subgraph of G by a (possibly empty) sequence of edge lifts. According to the last paper of the Graph Minor series [112], graphs are well-quasi-ordered under immersions, i.e., Theorem 1 holds also if we replace minors by immersions. Therefore, in order to prove a counterpart of Theorem 3 for the case of immersion-closed parameters, we need an algorithm that given an n-vertex graph G and a k-vertex graph H, checks where $H \leq_{im} G$ in $O_k \cdot (n(G))^3$ steps. Recently, a construction of such an algorithm was given in [61]. This makes it possible to derive the following meta-algorthmic result.

Theorem 4. *If* $\mathbf{p}$ *is a parameter that is closed under immersions, then there exists an algorithm that solves the problem* k-Parameter Checking for $\mathbf{p}$ *in* $f'(k) \cdot n^3$ *steps for some function* f'.

In fact, the main result of [61] proves the FPT membership of topological minor testing, i.e., given two graphs H and G, check whether $H \leq_t G$ (the parameter is the size of H). This means that there is a counterpart of Theorem 2 for the topological minor relation as well. This might create some hope that Theorem 3 holds for topological minors as well. Unfortunately, this requires an analogue of the combinatorial Theorem 1 which does not exist as it is possible to construct an infinite class of graphs that are pairwise non-comparable with respect to the topological minor relation: just take all cycles with their edges duplicated.

[5] Branchwidth was introduced in the paper X of the Graph Minor Series [106] and, from that point and then, was used as an alternative for treewidth (defined formally in Section 4). Treewidth and branchwidth can be seen as twin parameters, as the one is a constant factor approximation of the other.

An other argument for the non-existence of analogues of Theorems 1 and 3 for topological minors is given by the Topological Bandwidth problem asking whether the topological bandwidth of a graph is at most k. The topological bandwidth of a graph G is denoted by $\mathbf{tbw}(G)$ and is defined as

$$\mathbf{tbw}(G) = \min\{k \mid \exists q \geq 1 \colon G \leq_t P_q^k\}$$

(P_q^k is obtained by a path P_q of length q if we make adjacent any two vertices of distance $\leq k$ in P_q). It is easy to observe that $\mathbf{tbw}$ is closed under topological minors. In [42] it is mentioned that Topological Bandwidth is $\mathsf{W}[t]$-hard for all $t \geq 1$ – the proof is a modification of the proof for the case of Bandwidth in [15]. This implies that, under reasonable assumptions in parameterized complexity theory, the anti-chain corresponding to the $\leq_t$-obstruction family $\mathcal{O}_{\mathbf{tbw},k}^{\leq_t}$ is infinite for an infinite set of values of k.

4 Grid-Exclusion and Bidimensionality

4.1 Treewidth

Treewidth has been one of the main contributions of GMT to algorithmic graph theory. While, as a concept, its indices can be traced back to the work of Gavril in [56], its formal birth as a graph parameter occurred in the second paper of the Graph Minors series [104]. Currently, there are at least six equivalent definitions of tree-width. We present the original one from [104].

A *tree decomposition* of a graph G is a pair $(\mathcal{X}, T)$ where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

1. $\bigcup_{i \in V(T)} X_i = V(G)$;
2. for each edge $\{x, y\} \in E(G)$, $\{x, y\} \subseteq X_i$ for some $i \in V(T)$, and
3. for each $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} \{|X_i| - 1\}$. The *treewidth* of a graph G, denoted by $\mathbf{tw}(G)$, is the minimum width over all tree decompositions of G.

If, in the above definitions, we restrict the tree T to be a path then we define the notions of *path decomposition* and *pathwidth.* We write $\mathbf{pw}(G)$ to denote the pathwidth of a graph G. Pathwidth was defined earlier than treewidth in the first paper is the Graph Minors Series [102].

Treewidth can intuitively be seen as a measure of the topological resemblance of a graph to a tree or, alternatively, as a measure of the “global connectivity” of a graph. Similarly, pathwidth can be seen as a measure of the topological resemblance of a graph to a path.

Counting Monadic Second Order Logic (CMSOL) is a logic on graphs[6] where the domain is the set of vertices and edges, there are predicates for vertex-vertex

[6] We should stress that CMSOL is not only a logic on graphs but also on more general combinatorial objects called *strucures*.

adjacency and edge-vertex incidence, there is quantification over edges, vertices, edge sets and vertex sets, and there is a predicate $Card_{r,p}(S)$ which expresses whether the size of a set S is r modulo p.

The importance of treewidth for algorithmic graph theory is illustrated by the celebrated Courcelle's theorem stating that if Π_k is a parameterized property of graphs expressible by a CMSOL formula ϕ_k, then there is an algorithm that, given as input a graph G, can check whether G satisfies property Π_k (i.e., whether $G \in \Pi_k$) in $O_{|\phi_k|+\mathbf{tw}(G)}(n)$ steps. Moreover, there exists a meta-algorithm that, given ϕ_k, outputs such an algorithm. A proof of Courcelle's theorem can be found in [39, Chapter 6.5] and [44, Chapter 10] and similar results appeared by Arnborg, Lagergren, and Seese in [8] and Borie, Parker, and Tovey in [17]. An alternative game-theoretic proof has appeared recently in [81, 82].

Courcelle's theorem had a deep influence in parameterized algorithms as it automatically yields FPT-algoriths for a wide family of problems, provided that the treewidth of their instances is bounded by a function of the parameter k. The natural challenge is whether and when a parameterized problem can be reduced to its bounded treewidth variant. For this, an important step is to detect what kind of combinatorial structures are contained in a graph with big treewidth. The most prominent structure of this type is the grid G_k. Let $\mathbf{gm}(G)$ be the maximum k for which G contains G_k as a minor. A valuable theoretical tool in this direction was given by the following result of the GMT.

Theorem 5 ([105]). *There exists a recursive function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that $\mathbf{tw}(G) \leq f(\mathbf{gm}(G))$.*

While the above result appeared in the fifth paper of the series, a preliminary variant of it, where G is planar, appeared earlier in [103]. As every graph containing G_k as a minor has treewidth at least k, Theorem 5 implies that **tw** and **gm** are parametrically equivalent: a bound to the one of them implies a bound to the other. The initial estimation of the parameter dependence in Theorem 5 was huge. However, a better one appeared in [113] where it was proven that $\mathbf{tw}(G) = 20^{2\cdot(\mathbf{gm}(G))^5}$. An alternative, and relatively simpler, proof of Theorem 5 was given in [33]. To see the use of Theorem 5 in parameterized algorithm design, consider a parameter $\mathbf{p}$ that satisfies the following properties:

i. $\mathbf{p}$ is closed under taking of minors.
ii. there exists a recursive function $t : \mathbb{N} \rightarrow \mathbb{N}$ such that $\mathbf{p}(G_{t(k)}) > k$ for every non-negative integer k.
iii. One can construct an algorithm that, given a tree-decomposition of G of width at most q and an integer k, checks whether $\mathbf{p}(G) \leq k$ in $l(k,q) \cdot n^{O(1)}$ steps for some recursive function $l : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$.

Clearly, the first two conditions are easy to check for most instantiations of $\mathbf{p}$. Moreover, the third one follows directly from Courcelle's theorem if for each k, $\Pi_k = \{G \mid \mathbf{p}(G) \leq k\}$ is expressible by a CMSOL formula ϕ_k. There are many examples of such parameters. Typical examples are the vertex cover of a graph, i.e., the minimum number of vertices that meets all vertices of G and the feedback vertex set of a graph, i.e., the minimum number of vertices meeting all cycles of G. A direct consequence of Theorem 5 is the following (constructive) special case of Theorem 3:

Lemma 1. *Let* $\mathbf{p}$ *be a parameter satisfying conditions* **i**–**iii** *above for some* t *and* l. *Then it is possible to construct an algorithm that, given as input a graph* G *and an integer* k, *checks whether* $\mathbf{p}(G) \leq k$ *in* $(2^{O(f(t(k))} + l(k, 4 \cdot f(t(k)))) \cdot n^{O(1)}$ *steps where* f *is the function in Theorem 5.*

Proof. The algorithm in Lemma 1 works as follows: First of all, it uses an FPT-approximation algorithm for treewidth, i.e., an algorithm that given a graph G and an integer q, either outputs a tree decomposition of G of width at most $\alpha \cdot q$ or reports that $\mathbf{tw}(G) > q$ in $z(q) \cdot n^{\beta}$ steps. Various algorithms of this type have been proposed in [7, 14, 88, 100, 108] for different trade-offs between z, α, and β. Among them, we pick the one form [7] where $z(q) = 2^{4.38 \cdot q}$, $\alpha = 4$, and $\beta = 2$. We run this algorithm for G and $q = f(t(k))$. If it outputs a tree decomposition of width $\leq 4 \cdot f(t(k))$ then we use the algorithm of Property **iii** and solve the problem in $l(k, 4 \cdot f(t(k))) \cdot n^{O(1)}$ steps. If the algorithm reports that the treewidth of G is more than $f(t(k))$, then from Theorem 5, G contains $G_{t(k)}$ as a minor. In such a case, the algorithm directly outputs a negative answer as, from Properties **i** and **ii**, $\mathbf{p}(G) \geq \mathbf{p}(G_{t(k)}) > k$.

The idea of the above proof is also known as the *Win/win approach*: we either have an answer to the problems directly because the treewidth is big enough or we solve the problems use dynamic programming on a tree-decomposition of bounded width.

Clearly, the running time of the algorithm in Lemma 1 depends on the functions f, t, and l. In what follows, we comment on the current bounds on each one of them.

- l: As we have already mentioned, the (constructive) existence of l, follows from Courcelle's theorem for the wide family of problems that are expressible in CMSOL. However, the bounds on l, derived from the proof, are huge and this may dismiss any hope for a good parameter dependence (see [53]). However, for many problems it is possible to directly apply dynamic programing on the tree decomposition and derive moderate bounds on l such as $l(k, q) = 2^{O(q^2)} \cdot k^{O(1)}$, or $l(k, q) = 2^{O(q \log q)} \cdot k^{O(1)}$ or, even better, $l(k, q) = 2^{O(q)} \cdot k^{O(1)}$. Clearly, time bounds of the third type are more attractive. For this reason, we say that a parameter $\mathbf{p}$ is *single exponentially solvable with respect to treewidth* if there exists an algorithm that, given G and k, checks whether $\mathbf{p}(G) \leq k$ in $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ steps. There is a quite extended bibliography on how to do fast dynamic programming on graphs of bounded treewidth; as a sample of this, we just mention [5, 6, 9, 11, 13, 16, 22, 35, 35, 36, 37, 37, 38, 114, 120, 120].
- t: Bounds are much better for the function t. For most natural graph parameters, it holds that $t(k) = O(k)$ while for some of them, including $\mathbf{tw}$ and $\mathbf{pw}$, it holds that $t(k) = \Theta(k)$. However, there is a wide family of parameters

where $t(k) = O(\sqrt{k})$. This intuitively says that a certificate for the value of such a parameter spreads "bidimensionally" inside a $(k \times k)$-grid. For instance any vertex cover of G_k should have size at least $k \cdot \lfloor \frac{k}{2} \rfloor = \Omega(k^2)$ as the vertices of such a set should cover edges all over the "area" of the grid. Similarly, a feedback vertex set of G_k should have size at least $(\lfloor \frac{k}{2} \rfloor)^2 = \Omega(k^2)$ as the vertices of such a set should cover all $(\lfloor \frac{k}{2} \rfloor)^2$ members of a packing of "squares" in G_k.

If such a parameter is also closed under taking of minors then we call it *minor bidimensional.*

f: To improve the function f, i.e., the parameter dependence in Theorem 5, is an important challenge as, even for the parameters with most moderate instantiations of l and t, k-Parameter Checking for **p** could be only classified in $2^{2^{k^{O(1)}}} \cdot n^{O(1)}$-FPT. Robertson, Seymour, and Thomas conjectured in [113] that f can be a polynomial function. This would directly imply that k-Parameter Checking for **p** belongs to $2^{k^{O(1)}} \cdot n^{O(1)}$-FPT for a wide family of parameters (see [30] for more discussions and conjectures on this issue).

Another interesting problem is to lower bound the contribution of f in Theorem 5. As mentioned by Robertson, Seymour, and Thomas in [113] there are graphs excluding G_k as a minor that have treewidth $\Omega(k^2 \cdot \log k)$. To see this, one may use the result in [23] (see also [41, 122]) to construct an $O(1)$-regular Ramanujan graph G on n vertices that has girth $\Omega(\log n)$. One can easily verify that $\mathbf{gm}(G) = O(\frac{\sqrt{n}}{\log n})$. The claimed bound follows because Ramanujan graphs are expanders and thus $\mathbf{tw}(G) = \Omega(n)$. It is a challenging question whether any bound better than this one can be proven.

Towards achieving a polynomial dependance between treewidth and the size of an excluded grid, Reed and Wood defined in [101] the notion of a grid-like-minor. A *grid-like-minor* of order k in a graph G is a set of paths in G whose intersection graph is bipartite and contains a K_k as a minor. Clearly, the rows and columns of the $(k \times k)$-grid are a grid-like-minor of order $k+1$. In [101] it is proved that every graph with treewidth $\Omega(k^4 \sqrt{\log k})$ contains a grid-like minor of order k. Meta-algorithmic implications of the results in [101], analogous to those of Theorem 1, can be found in [85].

4.2 Bidimensionality

Theorem 5 has several refinements that are important for improving the parameter dependence of the algorithm in Lemma 1. The first variant of Theorem 5 for special graph classes appeared in [113] (proved for the twin parameter of branchwidth) from which it follows that if G is a planar graph, then $\mathbf{tw}(G) \leq 6 \cdot \mathbf{gm}(G)$. Actually, with some more careful application of the results of [113] it can also be proven that $\mathbf{tw}(G) \leq 5 \cdot \mathbf{gm}(G)$, which can be improved further to $\mathbf{tw}(G) \leq \frac{9}{2} \cdot \mathbf{gm}(G)$ using the results of [62]. An analogous upper bound holds also for graphs embedded in surfaces. From the results in [27], it follows that $\mathbf{tw}(G) \leq 6 \cdot (\mathbf{eg}(G) + 1) \cdot \mathbf{gm}(G)$ where $\mathbf{eg}(G)$ is the Euler genus of G. Also in [30], it was proven that if G is a $K_{3,r}$-minor free graph, then $\mathbf{tw}(G) \leq 20^{4r} \cdot \mathbf{gm}(G)$. At this point, the natural question is whether this linear dependence holds for every non-trivial minor-closed graph class. This was resolved in [29], where the following theorem has been proved.

Theorem 6. *Let r be a positive integer. If G is a K_r-minor free graph, then $\mathbf{tw}(G) = O_r(\mathbf{gm}(G))$.*

The proof of Theorem 6 is heavily based on GMT. More specifically, it depends on the Structure Theorem of the GMT [109] which implies immense bounds for the parameter dependence of the bound in Theorem 6. The improvement of the parameter dependance of Theorem 6 is an interesting problem and this might be possible without making use of the structural results of [109].

As mentioned in the previous Section, a parameter $\mathbf{p}$ is *minor-bidimensional* if it is closed under taking of minors and for every non-negative integer k it holds that $\mathbf{p}(G_{\lceil\sqrt{k}\rceil}) = \Omega(k)$. A major consequence of Lemma 1, Theorem 6, and the discussion above is the following meta-algorithmic result.

Theorem 7. *Let H be an r-vertex graph and let $\mathbf{p}$ be a graph parameter that is minor-bidimensional and single exponentially solvable with respect to treewidth. Then k-*PARAMETER CHECKING FOR $\mathbf{p}$ *restricted to H-minor free graphs belongs (constructively) to $2^{O_r(\sqrt{k})} \cdot n^{O(1)}$-*$\mathsf{FPT}$*, i.e., one can construct a sub-exponential* FPT*-algorithm that solves it.*

Notice that the above result is, in a sense, optimal, as, due to the complexity bounds in [18], a $2^{O(\sqrt{k})} \cdot n^{O(1)}$-step parameterized algorithm is the best we may expect for several bidimensional parameters, even on planar graphs. The meta-algorithmic machinery that we employed above in order to prove Theorem 7 is known as *Bidimensionality Theory* and was introduced for the first time in [27], while some preliminary ideas had already appeared in [4, 52].

Theorem 7 concerns only minor-closed parameters. A typical parameter that does not fit in the framework of minor-bidimensionality is the dominating set number, denoted by $\mathbf{ds}(G)$ and defined as the minimum size of a *dominating set* in G, i.e., a set S of vertices such that every vertex not in S has some neighbor in S.

The dominating set number is not minor-closed as it may increase by removing edges. However this is not the case when we do only contractions. To develop the contraction counterpart of bidimensionality, one has to find a counterpart of Theorem 6 for contractions, i.e., to detect what types of graphs appear as contractions in graphs with big treewidth. This line of research was developed in [26, 31] and concluded in [46]. Before we present the the results in [46], we need first some definitions.

Let Γ_k ($k \geq 2$) be the graph obtained from the $(k \times k)$-grid by triangulating internal faces of the $(k \times k)$-grid such that all internal vertices become of degree 6, all non-corner external vertices are of degree 4, and then one corner of degree two is joined by edges with all vertices of the external face (the *corners* are the vertices that in the underlying grid have degree two). Graph Γ_6 is shown

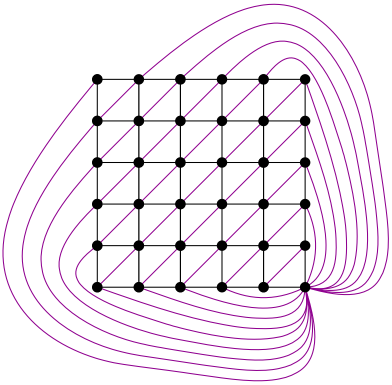

Fig. 3. The graph Γ_6

in Fig. 3. Let also Π_k be the graph obtained from Γ_k by adding a new vertex adjacent to all vertices of Γ_k.

A consequence of the results in [46] is the following.

Theorem 8. *There exists a function $\alpha : \mathbb{N} \to \mathbb{N}$ such that every connected graph of treewidth at least $\alpha(k)$ contains some of the graphs in $\{K_k, \Gamma_k, \Pi_k\}$ as a contraction.*

Theorem 8 has several refinements. One of them is the following counterpart of Theorem 6.

Theorem 9. *There exists a function $\beta : \mathbb{N} \to \mathbb{N}$ such that every connected K_r-minor-free graph of treewidth at least $\beta(r) \cdot k^2$ contains either Γ_k or Π_k as a contraction.*

Notice that in the above theorem, the quadratic dependence (on k) is optimal. Indeed, let Z_{k^2} be the graph obtained by adding to G_{k^2} a new vertex adjacent to all the k^2 vertices with both coordinates in the underlying grid divisible by k. Then Z_{k^2} excludes K_6, G_{k+2}, and Π_{k+2} as contractions and is of treewidth at least k^2. This means that, in order to have a "linear counterpart" of Theorem 6, we should restrict further the graphs that we exclude. An *apex graph* is a graph that can become planar by the removal of one vertex. It appears that the linear dependence in the bound of Theorem 6 is also possible for contractions when we consider graphs excluding some apex graph as a minor. For this, we define $\mathbf{tgm}(G)$ as the maximum k for which G contains Γ_k as a contraction.

Theorem 10. *Let H be an apex graph with r vertices. If G is a connected H-minor-free graph, then $\mathbf{tw}(G) = O_r(\mathbf{tgm}(G))$.*

We say that a parameter $\mathbf{p}$ is *contraction bidimensional* if it is closed under taking of contractions and if $\mathbf{p}(\Gamma_{\lceil\sqrt{k}\rceil}) = \Omega(k)$ for every non-negative integer

k. Using now Theorem 10 one can derive the following contraction counterpart of Theorem 7.

Theorem 11. *Let H be an r-vertex apex graph and let* $\mathbf{p}$ *be graph parameter that is contraction-bidimensional and single exponentially solvable with respect to treewidth. Then k-*Parameter Checking for $\mathbf{p}$ *restricted to H-minor free graphs belongs (constructively) to* $2^{O_r(\sqrt{k})} \cdot n^{O(1)}$-FPT*, i.e., one can construct a sub-exponential* FPT*-algorithm that solves it.*

The algorithmic consequence of Theorems 6 and 10 are not restricted in the design of sub-exponential parameterized algorithms (i.e., Theorems 7 and 11). Bidimensionality theory had meta-algorthmic applications in the automatic derivation of linear-time kernels for wide families of parameterized problems [12, 51]. Apart from its applications to parameterized complexity, Bidimensionality Theory was also used for the automated design of Fast Polynomial Time Approximation Schemes (FPTAS) in [28] and [48].

Proving extensions of Theorems 7 and 11 for wider families of graph classes (possibly with worse – but still moderate – time bounds) is a open challenge in parameterized algorithm design. For this, one may either need to find extensions of Theorems 6 and 10 for graph classes that are wider than H-minor free and apex-minor free graphs respectively (see [50] for an important step in this direction) or to invent alternative notions of grid-like structures whose presence in a graph is still able to certify a big enough value for the parameter $\mathbf{p}$ (see [85, 101] and the end of Subsection 4.1).

5 The Irrelevant Vertex Technique

One of the most powerful tools in parameterized algorithm design is the *irrelevant vertex technique*, introduced in [108] in order to derive (among others) FPT-algorithms for the H-Minor Checking (Theorem 2) and the k-Disjoint Paths Problem. The formal definition of the latter is the following.

k-Disjoint Paths
Instance: A graph G and a sequence of pairs
terminals $T = (s_1, t_1), \ldots, (s_k, t_k) \in (V(G) \times V(G))^k$.
Parameter: k.
Question: Are there k pairwise vertex disjoint paths
$P_1, \ldots, P_k$ in G such that for every $i \in \{1, \ldots, k\}$,
P_i has endpoints s_i and t_i?

We stress that, in [108], both H-Minor Checking and k-Disjoint Paths where treated simultaneously and the methodology that we present below is similar for both of them. In this section we give an outline of the $O_k(n^3)$ algorithm in [108] for the k-Disjoint Paths problem and we present some of the most important combinatorial results that supported the proof of its correctness.

5.1 The General Framework

Given an instance (G, T, k) of the k-Disjoint Paths problem, we say that a vertex $v \in V(G)$ is an *irrelevant* vertex of G if (G, T, k) and $(G \setminus v, T, k)$ are equivalent instances of the problem.

The general scheme of the algorithm in [108] is the following:

```
Irrelevant Vertex for the class $\mathcal{G}_k$
Input: An instance $(G, T, k)$ of $k$-Disjoint Paths
Output: A (reduced) equivalent instance of $k$-Disjoint Paths
1. while $G \notin \mathcal{G}_k$,
2.         find an irrelevant vertex $v$ in $G$
3.         set $G \leftarrow G \setminus v$
4. output $(G, T, k)$
```

Clearly, each variant of the above scheme depends on the parameterized class $\mathcal{G}_k$ and creates an equivalent instance that belongs to $\mathcal{G}_k$. The algorithm in [108] applies the above scheme in two *phases*: the first phase considers

$$\mathcal{G}_k = \{G \mid G \text{ is a } K_{h(k)}\text{-minor free graph}\}$$

for some recursive function h and produces equivalent instances where the input graph does not contain a "big clique" as a minor. The second phase assumes that the input graph excludes such a clique and considers

$$\mathcal{G}_k = \{G \mid G \text{ is a } G_{g(k)}\text{-minor free graph}\},$$

for some recursive function $g : \mathbb{N} \rightarrow \mathbb{N}$. This produces an equivalent instance that, from Theorem 5, has treewidth bounded by $O_k(1)$ and, in this case, the problem can by solved in $O_k(n)$ steps, using dynamic programming or, alternatively, by just using Courcelle's theorem.

It now remains to explain how Step 2 of the above scheme (i.e., finding an irrelevant vertex) is implemented in each of these two phases.

We omit the description of the first phase. Instead, we restrict ourselves to the second phase, as it encompasses the most combinatorially rich part of [108]. We just mention that the function h is determined from the results in [108] on the correctness of the first phase. The function g will be defined in the course of the description of the second phase below.

Assume now that we have an instance (G, T, k) of k-Disjoint Paths where G excludes a clique $K_{h(k)}$ as a minor but, however, it still contains a the grid $G_{g(k)}$ as a minor which means that $\mathbf{tw}(G) \geq g(k)$. A big part of [108] is devoted to the characterization of such graphs, i.e., of H-minor free graphs with "big" treewidth. In particular, a major achievement of [108] was to show the Weak Structure Theorem of GMT, stating that such graphs contain some portion that is, in a sense, "almost flat". At this point we postpone the description of the irrelevant vertex technique to Subsection 5.3 in order to give the precise statement of this theorem.

5.2 The Weak Structure Graph Minors Theorem

Walls. A *wall of height* k, $k \geq 1$, is the graph obtained from a $((k+1) \times (2 \cdot k+2))$-grid with vertices (x, y), $x \in \{1, \ldots, 2 \cdot k+2\}$, $y \in \{1, \ldots, k+1\}$, by the removal of the "vertical" edges $\{(x, y), (x, y+1)\}$ for odd $x+y$, and then the removal of all vertices of degree 1. We denote such a wall by W_k. The *corners* of the wall W_k are the vertices $c_1 = (1, 1)$, $c_2 = (2 \cdot k+1, 0)$, $c_3 = (2 \cdot k+1+(k+1 \mod 2), k+1)$ and $c_4 = (1+(k+1 \mod 2), k+1)$. We let $C = \{c_1, c_2, c_3, c_4\}$. A *subdivided wall W of height k* is a graph obtained from W_k by replacing some of its edges by paths without common internal vertices. We call the resulting graph W a *subdivision* of W_k. The *perimeter P* of a subdivided wall is the cycle defined by its boundary. The *layers* of a subdivided wall W of height k are recursively defined as follows. The first layer of W is its perimeter. For $i = 2, \cdots, \lfloor \frac{k}{2} \rfloor$, the i-th layer of W is the $(i-1)$-th layer of the subwall W' obtained from W by removing from W its perimeter and all occurring vertices of degree 1 (see Figure 4).

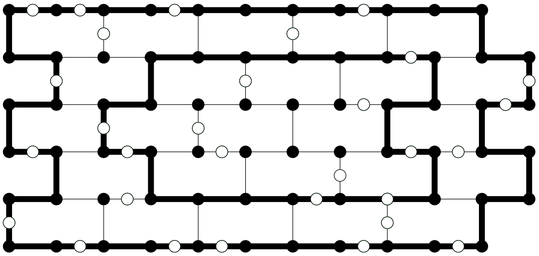

Fig. 4. A subdivided wall of height 5 and its two first layers. The first layer is its boundary

Compasses and Rural Divisions. Let W be a subdivided wall in G. Let K' be the connected component of $G \setminus P$ that contains $W \setminus P$. The *compass* K of W in G is the graph $G[V(K') \cup V(P)]$. Observe that W is a subgraph of K and K is connected. We say that a path of K is *perimetric* if its endpoints lie in the perimeter P of W. Let P_1 and P_2 be two perimetric paths of K with endpoints a_1, b_1 and a_2, b_2 respectively. We say that P_1 and P_2 *cross* in K if (a_1, a_2, b_1, b_2) is the cyclic ordering of their endpoints in P. We say that a wall is *flat* in G if K does not contain any pair of crossing and vertex-disjoint perimetric paths.

If J is a subgraph of K, we denote by $\partial_K J$ the set of all vertices $v \in V(J)$ such that either $v \in C$ or v is incident with an edge of K that is not in J. A *rural division* $\mathcal{D}$ of the compass K is a collection $(D_1, D_2, \ldots, D_m)$ of subgraphs of K with the following properties:

1. $\{E(D_1), E(D_2), \ldots, E(D_m)\}$ is a partition of non-empty subsets of $E(K)$,
2. for $i, j \in [m]$, if $i \neq j$ then $\partial_K D_i \neq \partial_K D_j$ and $V(D_i) \cap V(D_j) = \partial_K D_i \cap \partial_K D_j$,

3. for each $i \in [m]$ and all $x, y \in \partial_K D_i$ there exists a (x, y)-path in D_i with no internal vertex in $\partial_K D_i$,
4. for each $i \in [m]$, $|\partial_K D_i| \leq 3$, and
5. the hypergraph $H_K = (\bigcup_{i \in [m]} \partial_K D_i, \{\partial_K D_i \mid i \in [m]\})$ can be embedded in a closed disk Δ such that c_1, c_2, c_3 and c_4 appear in this order on the boundary of Δ and for each hyperedge e of H_K there exist $|e|$ mutually vertex-disjoint paths between e and C in K.

We call the elements of $\mathcal{D}$ *flaps*. A flap $D \in \mathcal{D}$ is *internal* if $V(D) \cap V(P) = \emptyset$. We can now state one of the main results in [108], known as the *Weak Structure Graph Minors theorem.*

Theorem 12 ([108]). *There exist recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every two graphs H and G and every $q \in \mathbb{N}$, one of the following holds:*

1. *H is a minor of G,*
2. *$\mathbf{tw}(G) \leq g_1(q, r)$, where $r = |V(H)|$*
3. *$\exists X \subseteq V(G)$ with $|X| \leq g_2(r)$ such that $G \setminus X$ contains as a subgraph a flat subdivided wall W where W has height q and the compass of W has a rural division $\mathcal{D}$ such that each internal flap of $\mathcal{D}$ has treewidth at most $g_1(r, q)$.*

While the statement of Theorem 12 above is somehow complicated, the intuition behind it is simpler. It says that when a graph excludes some "small" graph H as a minor and has "big enough" treewidth, it is enough to remove a "few" vertices from it, i.e., the vertices in X, and take a graph $G \setminus X$ where it is possible to detect a subdivided wall W that is situated in a "flat" territory inside its perimeter P. The part of G that is inside P is the compass K of W which can be seen as the union of a collection of graphs (flaps) that are tree-like (have bounded treewidth) and are "planted" in that territory. Theorem 12 was used also in [2, 24] with the name "the Trinity Lemma". However, a more depictive alternative nomenclature might be the "Sunny Forest Lemma", in the sense that the compass K is a forest, whose trees are the flaps, and X is the sun throwing its rays at it!

In [59], an optimized version of the above result was proved where $g_1(r, q) = O_r(q)$ and $g_2(r)$ is equal to the apex number of H, i.e., the minimum number of vertices that, when removed from H, leave a planar graph. This improved version can easily yield both Theorems 6 and 10. In case H is an apex graph, i.e., it can become planar with the removal of a single vertex, the result in [59] implies that $X = \emptyset$ which gives an analogue of Theorem 12 for apex minor-free graphs. As, in this case, the "sun" X does not exist, we are tempted to call this "apex"-variant of Theorem 12 the "Dark Forest Lemma".

5.3 Irrelevant Vertices and Linkages

We now go back to the task of detecting an irrelevant vertex in a graph G that excludes $K_{h(k)}$ and has treewidth bigger than $g(k)$. Recall that, at this point, h

has already been determined so that the previous phase of the algorithm runs correctly. In what follows, we set $g(k) = g_1(f_0(k) \cdot f_1(\lambda(k)), h(k))$ where g_1 is the function in Theorem 12, $f_0(k) = \lceil\sqrt{2k}\rceil + 1$, and f_1 and λ will be determined later.

According to [108], it is possible, in $O_k(n^2)$ steps, to detect in G a set X and a subdivided wall W of height $q = f_0(k) \cdot f_1(\lambda(k))$ of $G \setminus X$ where $|X| \leq g_2(h(k))$, as indicated in Theorem 12. For simplicity, we restrict our presentation to the case where X is an empty set, i.e., $|X| = 0$. Even if the ideas for the more general case are of the same flavor, they are quite more complicated and we prefer to omit them here.

Using a counting argument based on the definition of f_0, it is easy to see that W contains a subdivided wall W' of height $q' = f_1(\lambda(k))$ whose compass K' avoids all terminals of the pairs in T.

The next step of the algorithm in [108] is based on the claim that if we take q' to be "big enough", then any vertex v_{mid} of the *inner* layer L_{in} of W' is an irrelevant vertex and therefore it can be safely removed from G. While such a vertex is easy to detect, to proof that it is indeed irrelevant – for some suitable choice of q' – is not easy. We just mention that papers XXI [111] and XXII [107] of the Graph Minor series where devoted to it. Below, we present only some basic notions and ideas used in this proof. For this, we first need the definition of a k-linkage, introduced in [111].

A *k-linkage* in a graph G is a set of k pairwise disjoint paths of it. The *endpoints* of a linkage $\mathcal{L}$ are the endpoints of the paths in $\mathcal{L}$. The *pattern* of $\mathcal{L}$ is defined as

$$\pi(\mathcal{L}) = \{\{s,t\} \mid \mathcal{L} \text{ contains a path from } s \text{ to } t\}$$

Two k-linkages are *equivalent* if they have the same pattern.

W.l.o.g. we assume that all terminals involved in T are distinct. This implies that every solution to the k-Disjoint Paths problem is a k-linkage, whose pattern is determined by the pairs in T. To prove the irrelevance of the vertex v_{mid}, it is enough to show that any linkage $\mathcal{L}$ whose paths meet L_{in} can be replaced with an equivalent one that avoids it. To obtain an idea of how paths in $\mathcal{L}$ may reside inside K', we need to make some observations.

Let $\mathcal{R}$ be the linkage defined by the connected components of $(\bigcup_{L \in \mathcal{L}} L) \cap K'$, i.e., the subpaths of the paths in $\mathcal{L}$ that are "cropped" by the compass K' (notice that all paths in $\mathcal{R}$ are perimetric). By the flatness of W', it is not possible that two paths in $\mathcal{R}$ cross in K'. Moreover, by the definition of the the rural division $\mathcal{D}'$ of K', each layer of W', different than the inner one, is a separator of G. Therefore, if a path in $\mathcal{R}$ meets layers L_i and L_j for $i \leq j$, then it should also meet layer L_μ for every $\mu \in \{i, \ldots, j\}$. These observations argue that, intuitively, paths in $\mathcal{R}$ cross K' as if K' where a graph embedded in a disk bounded by P – see Figure 5 for a visualization of this. One may now claim that the infrastructure of a "big enough" subdivided wall W' should provide enough space inside K'

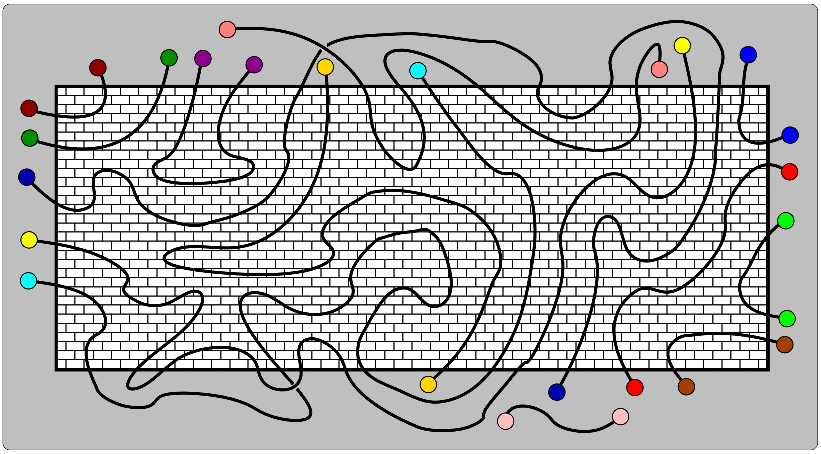

Fig. 5. A subdivided wall W' and the way a 13-linkage $\mathcal{L}$ is traversing its compass K'. The only vertices that are depicted are the endpoints of the paths in $\mathcal{L}$ (white vertices). The only edges that are depicted are those of the paths in $\mathcal{L}$ and the edges of W'. The grey area contains the vertices and the edges of the graph G that do not belong to K'.

so that the paths of $\mathcal{L}$ could be rerouted to an equivalent linkage that does not enter very deeply inside K'. To formalize this claim Robertson and Seymour defined the notion of a vital linkage in [111].

A linkage $\mathcal{L}$ in a graph G is called *vital* if its vertices meet all the vertices of G and if there is no other linkage in G that is equivalent to $\mathcal{L}$. An example of a vital k-linkage in a graph is depicted in Figure 6. Clearly, if a solution of the k-Disjoint Paths Problem corresponds to a vital linkage, then no irrelevant vertex can be detected. The main result of [111] asserts that this possible "lack of flexibility" of linkages vanishes when graphs have big enough treewidth.

Theorem 13. *There exists a recursive function $\lambda : \mathbb{N} \rightarrow \mathbb{N}$ such that every graph with a vital k-linkage has treewidth at most $\lambda(k)$.*

Actually, it was also proved in [111] that treewidth can be replaced by pathwidth in Theorem 13. As the proof of 13 uses the Structure Theorem of the GMT [109], the upper bound for λ that follows from [111] is immense. However it was proved in [3] that in the case of planar graphs it holds that $\lambda(k) = 2^{O(k)}$. Moreover, this bound is, in a sense, tight: as argued in [3], for each k it is possible to construct a planar graph that contains a vital k-linkage and has treewidth $2^{\Omega(k)}$ (the 5-linkage in the graph of Figure 6 already gives the flavor of such a construction).

Let now G' be the subgraph of G defined by the union of the paths in $\mathcal{L}$, and the compass K' of W'. At this point, a naive idea might be to directly apply Theorem 13 and set $q' = \lambda(k)$ so that the linkage $\mathcal{L}$ of G', corresponding to a solution of the k-Disjoint Paths problem, cannot be vital. However, from this alone, we cannot expect nothing better than avoiding some vertices that will not

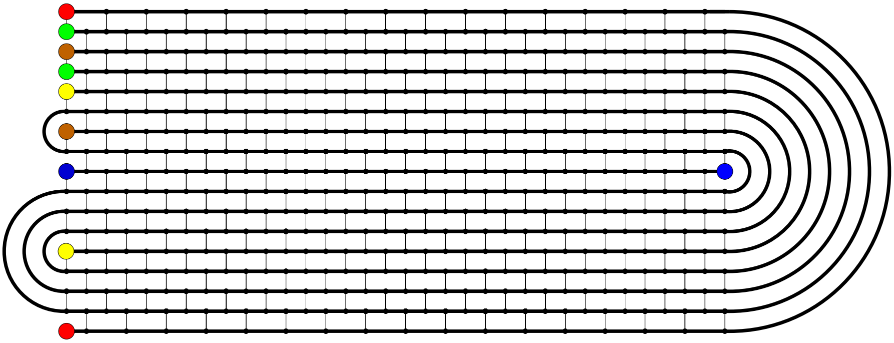

Fig. 6. A graph of treewidth 17 and a vital 5-linkage in it

necessarily be the vertices in L_{in}. Therefore, a non-vital linkage alone does not provide the flexibility we need in order to reroute in G' the paths of $\mathcal{L}$ in a way that L_{in} is avoided.

Curiously, it appears that the importance of non-vital linkages is rather qualitative than quantitative. Based on their "elementary" flexibility, it is possible to prove that none of the paths in $\mathcal{R}$ "bounces" much. In particular, $\mathcal{L}$ can be chosen in a way that if a path in $\mathcal{R}$ meets some layer L_i in two different vertices x and y, then its subpath between x and y will not meet any layer L_j where $|i - j| \geq f_1(\lambda(k))$, for some recursive function f_1. This directly implies that paths in $\mathcal{R}$ do not go deeper than layer $L_{f_1(\lambda(k))}$ and thus they avoid L_{in} when $q' = f_1(\lambda(k))$. That way, it is possible to prove what we need: if the height of W' is $f_1(\lambda(k))$, then another linkage, equivalent to $\mathcal{L}$ exists in G' (and therefore in G as well) that avoids L_{in}.

We should stress that even if the above sketch might be "convincing" for a good-tempered reader, it is far from being a formal proof. In the more realistic case where X is non-empty, a more complicated criterion for the choice of the subdivided wall W' should be devised and a bigger lower bound for the height of W' is necessary so that it contains an irrelevant vertex. In fact, this requires bigger lower bounds for both f_0 and f_1. The whole proof is quite technical and has been the main purpose of [107].

According to the above discussion, the second phase of the algorithm runs in $O_k(n^3)$ steps and outputs a graph of treewidth at most $g(k)$. As proved in [116], the k-Disjoint Paths problem can be solved by a $f_2(k)\cdot n$ step dynamic programming algorithm where $f_2(k) = 2^{O(k \log k)}$ (see [1, 90] for results related to this problem). As the parameter dependence of the running time of this last step is dominant in the running time of the algorithm, we conclude that the overall parameter dependence is:

$$O(f_2(g_1(f_0(k) \cdot f_1(\lambda(k)), h(k)))).$$

Clearly, an improvement on the existing bounds for any of the functions $g_1, h, f_0, f_1,$ f_2, and λ would be an important step towards reducing the parameter dependence of the algorithm for the k-Disjoint Paths problem. In fact, the only function that is "really immense" is λ, because the proof of its existence was based on the Structure Theorem of the GMT [109]. In this direction an alternative, relatively simpler, proof was given in [80] that avoids the core results of [109]. Using a rough estimation, the proof in [80] should give that that $\lambda(k) = 2^{2^{2^{O(k)}}}$ which changes the status of the parameter dependence in Roberson and Seymour's algorithm from "immense" to "huge". Clearly, any further improvements, even for special cases or variants of the problem, are highly welcome (see [3]).

5.4 Applications

The above description already outlines a powerful algorithmic framework that could not be of use for just one problem. Below, we mention a series of results in parameterized algorithms where the irrelevant vertex technique (or extensions of it) has been applied. We sort them in chronological order of their appearance.

[24] A proof of the following meta-agorithmic result: *Let $\mathcal{C}$ be a class of graphs excluding and h-vertex graph H as a minor. Then any first-order definable decision problem can be solved in time $O_{h+|\phi|}(n^{O(1)})$, where f is a computable function and ϕ is the sentence defining the decision problem.*

[77] A $2^{O(g)} \cdot n$ step algorithm that, given a graph G and a non-negative integer g either outputs an embedding of G in a surface of genus g or a minor of G that belongs to $\mathbf{obs}_{\leq_m}(\mathcal{G}_g)$ where $\mathcal{G}_g$ contains all graphs embeddible in a surface of Euler genus g. A previous result of this type, but not with single-exponential parameter dependence appeared previously in [94].

[2] A proof that it is possible to construct an algorithm that, given the obstruction sets of two minor-closed graph classes $\mathcal{G}_1$ and $\mathcal{G}_2$, outputs the obstruction set of the class $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$. Also, in the same paper, it was proved that it is possible to construct an algorithm that given the obstruction set of a minor-closed graph class $\mathcal{G}$ and a non-negative integer k, outputs the obstruction set of the class k-$\mathbf{almost}(\mathcal{G}) = \{G \mid \exists S \subseteq V(G) : |S| \leq k \text{ and } G \setminus S \in \mathcal{G}\}$ (see also [19, 71, 93] for related results).

[83] An $O_k(n^{O(1)})$ time algorithm for solving the Induced Cycle Through Terminals problem: *Given a graph G, embedded in some surface, and a set $S \subseteq V(G)$ of terminals, does G contain an induced cycle that meets all vertices in S?*

[60] An $2^{O(k^{3/2})} \cdot n^{O(1)}$ time algorithm for the Odd Induced Cycle Packing on planar graphs: *Given a graph G and an integer k, does G contains k induced odd cycles?*

[79] An $O_k(n^{O(1)})$ time algorithm for the Odd Cycle Packing problem: *Given a graph G and an integer k, does G contains k vertex-disjoint odd cycles?*

[64] An $O_k(n)$ time algorithm for the BIPARTITE CONTRACTION problem: *Given a graph G and an integer k, can we obtain a bipartite graph from G by a sequence of at most k edge contractions in G?*

[49] Subexponential $2^{O(\sqrt{k})} \cdot n$ time algorithms for the PARTIAL VERTEX COVER and PARTIAL DOMINATING SET problems for apex minor-free graphs: *Given a graph G and integers k, t, can we cover* (resp. *dominate*) *at least t edges (resp. vertices) with at most k vertices?*

[61] An $O_k(n^3)$ algorithm for the TOPOLOGICAL MINOR CONTAINMENT and the IMMERSION CONTAINMENT problems: *Given two graphs G and H, where $n(H) = k$, does G contain H as a topological minor* (resp. *immersion*). The results in [61] can be seen as a major extension of the algorithm in [108].

[45] A proof of the following result on kernelization: *Let $\mathcal{G}_r$ be the class of all K_r-minor free graphs. Then the* DOMINATING SET *Problem and the* CONNECTED DOMINATING SET *problem, asking whether a graph G has a (connected) dominating set of size k, has a linear $O_r(k)$-size kernel when restricted in graphs in $\mathcal{G}_r$.*

[69] An $O_{k+g}(n^3)$ algorithm for the CONTRACTION CONTAINMENT problem restricted to graphs of Euler genus g. The CONTRACTION CONTAINMENT problem asks, with input two graphs G and H, where $n(H) = k$, whether H is a contraction of G.

Clearly, the above list is just indicative and is expected to grow more. Further algorithmic applications of the weak structure theorem and/or the irrelevant vertex technique can be found in [66, 68, 70, 72, 73, 75, 76, 78]. Also results where the irrelevant vertex idea is applied an a more general sense, without using directly results of the GMT, can be found in [25, 47, 63, 92].

6 Conclusions

Covering the whole range of the contributions of the GMT to the design of parameterized algorithms is a task that cannot fit in the space of this short presentation. The progress over the last years towards building an Algorithmic Graph Minors Theory has been noticeable and we believe that there is much more "algorithmic material" to be extracted from this deep and fascinating theory. As we expect more results to emerge from GMT, not only in parameterized algorithms by also in other fields of algorithm design, we hope that this small portion of the material covered will be of use as in invitation to this direction.

Acknowledgements. We wish to thank Isolde Adler, Marcin Kamiński and Stavros G. Kolliopoulos for their detailed comments, remarks, and suggestions on this text. We also wish to personally thank Michael R. Fellows for his support and encouragement while investigating this "minor" corner of parameterized algorithms.

References

1. Adler, I., Dorn, F., Fomin, F.V., Sau, I., Thilikos, D.M.: Fast Minor Testing in Planar Graphs. In: de Berg, M., Meyer, U. (eds.) ESA 2010. LNCS, vol. 6346, pp. 97–109. Springer, Heidelberg (2010)
2. Adler, I., Grohe, M., Kreutzer, S.: Computing excluded minors. In: Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, pp. 641–650. Society for Industrial and Applied Mathematics, Philadelphia (2008)
3. Adler, I., Kolliopoulos, S.G., Krause, P.K., Lokshtanov, D., Saurabh, S., Thilikos, D.: Tight Bounds for Linkages in Planar Graphs. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 110–121. Springer, Heidelberg (2011)
4. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica 33(4), 461–493 (2002)
5. Alber, J., Dorn, F., Niedermeier, R.: Experimental evaluation of a tree decomposition-based algorithm for vertex cover on planar graphs. Discrete Applied Mathematics 145(2), 219–231 (2005); Structural Decompositions, Width Parameters, and Graph Labelings
6. Alber, J., Niedermeier, R.: Improved Tree Decomposition Based Algorithms for Domination-like Problems. In: Rajsbaum, S. (ed.) LATIN 2002. LNCS, vol. 2286, pp. 613–627. Springer, Heidelberg (2002)
7. Amir, E.: Approximation algorithms for treewidth. Algorithmica 56, 448–479 (2010)
8. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. Journal of Algorithms 12, 308–340 (1991)
9. Betzler, N., Niedermeier, R., Uhlmann, J.: Tree decompositions of graphs: Saving memory in dynamic programming. Discrete Optimization 3(3), 220–229 (2006); Graphs and Combinatorial Optimization
10. Bienstock, D., Dean, N.: On obstructions to small face covers in planar graphs. J. Combin. Theory Ser. B 55(2), 163–189 (1992)
11. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Fourier meets Möbius: fast subset convolution. In: STOC, pp. 67–74. ACM (2007)
12. Bodlaender, H., Fomin, F., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.: (Meta) kernelization. In: 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009). ACM (2009)
13. Bodlaender, H.L.: Dynamic Programming on Graphs with Bounded Treewidth. In: Lepistö, T., Salomaa, A. (eds.) ICALP 1988. LNCS, vol. 317, pp. 105–118. Springer, Heidelberg (1988)
14. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. 25(6), 1305–1317 (1996)
15. Bodlaender, H.L., Fellows, M.R., Hallett, M.T.: Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy. In: Twenty-sixth Annual ACM Symposium on Theory of Computing (STOC 1994), pp. 449–458. ACM, New York (1994)
16. Bodlaender, H.L., Telle, J.A.: Space-efficient construction variants of dynamic programming. Nordic J. Comput. 11(4), 374–385 (2004)
17. Borie, R.B., Parker, R.G., Tovey, C.A.: Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. Algorithmica 7, 555–581 (1992)

18. Cai, L., Juedes, D.: On the existence of subexponential parameterized algorithms. J. Comput. System Sci. 67(4), 789–807 (2003)
19. Cattell, K., Dinneen, M.J., Downey, R.G., Fellows, M.R., Langston, M.A.: On computing graph minor obstruction sets. Theor. Comput. Sci. 233, 107–127 (2000)
20. Chlebíková, J.: The structure of obstructions to treewidth and pathwidth. Discrete Applied Mathematics 120(1-3), 61–71 (2002)
21. Courcelle, B., Downey, R.G., Fellows, M.R.: A note on the computability of graph minor obstruction sets for monadic second order ideals. In: First Japan-New Zealand Workshop on Logic in Computer Science, Auckland, vol. 3, pp. 1194–1198 (1997) (electronic)
22. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time. In: IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011), pp. 150–159. IEEE Computer Society (2011)
23. Dahan, X., Tillich, J.-P.: Ramanujan graphs of very large girth based on octonions. CoRR, arXiv:1011.2642 (November 2010-2011)
24. Dawar, A., Grohe, M., Kreutzer, S.: Locally excluding a minor. In: 21st IEEE Symposium on Logic in Computer Science (LICS 2007), pp. 270–279. IEEE, New York (2007)
25. Dawar, A., Kreutzer, S.: Domination problems in nowhere-dense classes. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FST-TCS 2009), pp. 157–168 (2009)
26. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Bidimensional parameters and local treewidth. SIAM J. Discrete Math. 18(3), 501–511 (2004) (electronic)
27. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. Journal of the ACM 52(6), 866–893 (2005)
28. Demaine, E.D., Hajiaghayi, M.: Bidimensionality: new connections between FPT algorithms and PTASs. In: Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 590–601. ACM, New York (2005) (electronic)
29. Demaine, E.D., Hajiaghayi, M.: Linearity of grid minors in treewidth with applications through bidimensionality. Combinatorica 28(1), 19–36 (2008)
30. Demaine, E.D., Hajiaghayi, M., Kawarabayashi, K.: Algorithmic graph minor theory: Improved grid minor bounds and Wagner's contraction. Algorithmica 54(2), 142–180 (2009)
31. Demaine, E.D., Hajiaghayi, M., Thilikos, D.M.: The bidimensional theory of bounded-genus graphs. SIAM J. Discrete Math. 20(2), 357–371 (2006)
32. Diestel, R.: Graph Theory, 3rd edn. Graduate Texts in Mathematics, vol. 173. Springer (2005)
33. Diestel, R., Jensen, T.R., Gorbunov, K.Y., Thomassen, C.: Highly connected sets and the excluded grid theorem. J. Combin. Theory Ser. B 75(1), 61–73 (1999)
34. Dinneen, M.J.: Too many minor order obstructions (for parameterized lower ideals). In: First Japan-New Zealand Workshop on Logic in Computer Science, Auckland, vol. 3(11), pp. 1199–1206 (1997) (electronic)
35. Dorn, F.: Dynamic Programming and Fast Matrix Multiplication. In: Azar, Y., Erlebach, T. (eds.) ESA 2006. LNCS, vol. 4168, pp. 280–291. Springer, Heidelberg (2006)
36. Dorn, F., Fomin, F.V., Thilikos, D.M.: Fast Subexponential Algorithm for Non-local Problems on Graphs of Bounded Genus. In: Arge, L., Freivalds, R. (eds.) SWAT 2006. LNCS, vol. 4059, pp. 172–183. Springer, Heidelberg (2006)

37. Dorn, F., Fomin, F.V., Thilikos, D.M.: Catalan structures and dynamic programming in H-minor-free graphs. In: ACM-SIAM Symposium on Discrete Algorithms (SODA 2008), pp. 631–640. SIAM (2008)
38. Dorn, F., Penninkx, E., Bodlaender, H.L., Fomin, F.V.: Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. Algorithmica 58(3), 790–810 (2010)
39. Downey, R.G., Fellows, M.R.: Parameterized complexity. Monographs in Computer Science. Springer, New York (1999)
40. Ellis, J.A., Sudborough, I.H., Turner, J.S.: The vertex separation and search number of a graph. Information and Computation 113(1), 50–79 (1994)
41. Erdős, P., Sachs, H.: Reguläre graphen gegebener tailenweite mit minimaler knollenzahh. Wiss. Z. Univ. Halle-Willenberg Math. Nat. 12, 251–258 (1063)
42. Fellows, M.: Towards Fully Multivariate Algorithmics: Some New Results and Directions in Parameter Ecology. In: Fiala, J., Kratochvíl, J., Miller, M. (eds.) IWOCA 2009. LNCS, vol. 5874, pp. 2–10. Springer, Heidelberg (2009)
43. Fellows, M.R., Langston, M.A.: On search, decision, and the efficiency of polynomial-time algorithms. J. Comput. System Sci. 49(3), 769–779 (1994)
44. Flum, J., Grohe, M.: Parameterized Complexity theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)
45. Fomin, F.V., Daniel Lokshtanov, S.S., Thilikos, D.M.: Linear kernels for (connected) dominating set on h-minor-free graphs. In: 23st ACM–SIAM Symposium on Discrete Algorithms (SODA 2012). ACM-SIAM, San Francisco (2012)
46. Fomin, F.V., Golovach, P.A., Thilikos, D.M.: Contraction obstructions for treewidth. J. Comb. Theory, Ser. B 101(5), 302–314 (2011)
47. Fomin, F.V., Lokshtanov, D., Misra, N., Philip, G., Saurabh, S.: Hitting forbidden minors: Approximation and kernelization. In: STACS, pp. 189–200 (2011)
48. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S.: Bidimensionality and EPTAS. In: 22st ACM–SIAM Symposium on Discrete Algorithms (SODA 2011), pp. 748–759. ACM-SIAM, San Francisco (2011)
49. Fomin, F.V., Lokshtanov, D., Raman, V., Saurabh, S.: Subexponential algorithms for partial cover problems. Inf. Process. Lett. 111(16), 814–818 (2011)
50. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Bidimensionality and geometric graphs. In: 23st ACM–SIAM Symposium on Discrete Algorithms (SODA 2012). ACM-SIAM, San Francisco (2012)
51. Fomin, F.V., Lokshtanov, D., Saurabh, S., Thilikos, D.M.: Bidimensionality and kernels. In: 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010), Austin, Texas, pp. 503–510. ACM-SIAM (2010)
52. Fomin, F.V., Thilikos, D.M.: Dominating sets in planar graphs: branch-width and exponential speed-up. SIAM J. Comput. 36(2), 281–309 (2006) (electronic)
53. Frick, M., Grohe, M.: The complexity of first-order and monadic second-order logic revisited. Ann. Pure Appl. Logic 130(1-3), 3–31 (2004)
54. Friedman, H., Robertson, N., Seymour, P.D.: The metamathematics of the graph minor theorem. In: Logic and Combinatorics (Arcata, Calif., 1985). Contemp. Math., vol. 65, pp. 229–261. Amer. Math. Soc., Providence (1987)
55. Friedman, H.M.: Internal finite tree embeddings. In: Reflections on the foundations of mathematics (Stanford, CA, 1998). Lect. Notes Log., vol. 15, pp. 60–91. Assoc. Symbol. Logic, Urbana (2002)
56. Gavril, F.: The intersection graphs of subtrees in trees are exactly the chordal graphs. J. Combin. Theory, Ser. B 16(1), 47–56 (1974)

57. Geelen, J.F., Gerards, A.M.H., Robertson, N., Whittle, G.P.: On the excluded minors for the matroids of branch-width k. J. Combin. Theory Ser. B 88(2), 261–265 (2003)
58. Giannopoulou, A.C., Thilikos, D.M.: Obstructions for tree-depth. Electronic Notes in Discrete Mathematics 34, 249–253 (2009)
59. Giannopoulou, A.C., Thilikos, D.M.: Optimizing the graph minors weak structure theorem. CoRR, arXiv:1102.5762 (February 2011)
60. Golovach, P.A., Kamiński, M., Paulusma, D., Thilikos, D.M.: Induced Packing of Odd Cycles in a Planar Graph. In: Dong, Y., Du, D.-Z., Ibarra, O. (eds.) ISAAC 2009. LNCS, vol. 5878, pp. 514–523. Springer, Heidelberg (2009)
61. Grohe, M., Kawarabayashi, K., Marx, D., Wollan, P.: Finding topological subgraphs is fixed-parameter tractable. In: Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC 2011), pp. 479–488 (2011)
62. Gu, Q.-P., Tamaki, H.: Improved Bounds on the Planar Branchwidth with Respect to the Largest Grid Minor Size. In: Cheong, O., Chwa, K.-Y., Park, K. (eds.) ISAAC 2010, Part II. LNCS, vol. 6507, pp. 85–96. Springer, Heidelberg (2010)
63. Heggernes, P., van't Hof, P., Jansen, B.M.P., Kratsch, S., Villanger, Y.: Parameterized Complexity of Vertex Deletion into Perfect Graph Classes. In: Owe, O., Steffen, M., Telle, J.A. (eds.) FCT 2011. LNCS, vol. 6914, pp. 240–251. Springer, Heidelberg (2011)
64. Heggernes, P., van't Hof, P., Lokshtanov, D., Paul, C.: Obtaining a bipartite graph by contracting few edges. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011), pp. 217–228 (2011)
65. Higman, G.: Ordering by divisibility in abstract algebras. Proc. London Math. Soc. 2(3), 326–336 (1952)
66. Ito, T., Kamiński, M., Paulusma, D., Thilikos, D.M.: Parameterizing cut sets in a graph by the number of their components. Theor. Comput. Sci. 412(45), 6340–6350 (2011)
67. Johnson, D.S.: The NP-completeness column: An ongoing guide. Journal of Algorithms 8(2), 285–303 (1987)
68. Kamiński, M., Nishimura, N.: Finding an induced path of given parity in planar graphs in polynomial time. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011), pp. 656–670. ACM (2012)
69. Kamiński, M., Thilikos, D.M.: Contraction checking in graphs on surfaces. In: 29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012), pp. 182–193 (2012)
70. Kawarabayashi, K.: Half integral packing, Erdős-Pósa-property and graph minors. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 1187–1196. Society for Industrial and Applied Mathematics, Philadelphia (2007)
71. Kawarabayashi, K.: Planarity allowing few error vertices in linear time. In: 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, pp. 639–648 (2009)
72. Kawarabayashi, K., Kobayashi, Y.: The edge disjoint paths problem in eulerian graphs and 4-edge-connected graphs. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, pp. 345–353. Society for Industrial and Applied Mathematics, Philadelphia (2010)
73. Kawarabayashi, K., Kobayashi, Y.: An improved algorithm for the half-disjoint paths problem. SIAM J. Discrete Math. 25(3), 1322–1330 (2011)

74. Kawarabayashi, K., Kobayashi, Y., Reed, B.: The disjoint paths problem in quadratic time. Journal of Combinatorial Theory, Series B (2011)
75. Kawarabayashi, K., Kreutzer, S., Mohar, B.: Linkless and flat embeddings in 3-space and the unknot problem. In: Proceedings of the 2010 Annual Symposium on Computational Geometry, SoCG 2010, pp. 97–106. ACM, New York (2010)
76. Kawarabayashi, K., Li, Z., Reed, B.A.: Recognizing a totally odd K_4-subdivision, parity 2-disjoint rooted paths and a parity cycle through specified elements. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, pp. 318–328 (2010)
77. Kawarabayashi, K., Mohar, B., Reed, B.A.: A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, pp. 771–780 (2008)
78. Kawarabayashi, K., Reed, B.A.: Hadwiger's conjecture is decidable. In: 41st Annual ACM Symposium on Theory of Computing (STOC 2009), pp. 445–454 (2009)
79. Kawarabayashi, K., Reed, B.A.: Odd cycle packing. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, pp. 695–704 (2010)
80. Kawarabayashi, K., Wollan, P.: A shorter proof of the graph minor algorithm: the unique linkage theorem. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, pp. 771–780 (2008)
81. Kneis, J., Langer, A.: A practical approach to courcelle's theorem. Electron. Notes Theor. Comput. Sci. 251, 65–81 (2009)
82. Kneis, J., Langer, A., Rossmanith, P.: Courcelle's theorem - a game-theoretic approach. CoRR, arXiv:1104.3905 (April 2011)
83. Kobayashi, Y., Kawarabayashi, K.: Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In: 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2009), pp. 1146–1155. ACM-SIAM (2009)
84. Koutsonas, A., Thilikos, D.M., Yamazaki, K.: Outerplanar obstructions for matroid pathwidth. In: EuroComb 2011: European Conference on Combinatorics, Graph Theory and Applications (2011)
85. Kreutzer, S., Tazari, S.: On brambles, grid-like minors, and parameterized intractability of monadic second-order logic. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, pp. 354–364. Society for Industrial and Applied Mathematics, Philadelphia (2010)
86. Kruskal, J.B.: Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture. Trans. Amer. Math. Soc. 95, 210–225 (1960)
87. Kuratowski, K.: Sur le problème des courbes gauches en topologie. Fund. Math. 15, 271–283 (1930)
88. Lagergren, J.: Efficient parallel algorithms for graphs of bounded tree-width. Journal of Algorithms 20(1), 20–44 (1996)
89. Lagergren, J.: Upper bounds on the size of obstructions and intertwines. J. Combin. Theory, Ser. B 73, 7–40 (1998)
90. Lokshtanov, D., Marx, D., Saurabh, S.: Slightly superexponential parameterized problems. In: 22st ACM–SIAM Symposium on Discrete Algorithms (SODA 2011), pp. 760–776 (2011)
91. Lovász, L.: Graph minor theory. Bull. Amer. Math. Soc. (N.S.) 43(1), 75–86 (2006) (electronic)
92. Marx, D.: Chordal deletion is fixed-parameter tractable. Algorithmica 57(4), 747–768 (2010)
93. Marx, D., Schlotter, I.: Obtaining a planar graph by vertex deletion. Algorithmica 62(3-4), 807–822 (2012)

94. Mohar, B.: A linear time algorithm for embedding graphs in an arbitrary surface. SIAM J. Discrete Math. 12(1), 6–26 (1999)
95. Nash-Williams, C.S.J.A.: On well-quasi-ordering finite trees. Proc. Cambridge Philos. Soc. 59, 833–835 (1963)
96. Neumann, B.H.: On ordered division rings. Trans. Amer. Math. Soc. 66, 202–252 (1949)
97. Niedermeier, R.: Invitation to fixed-parameter algorithms. Oxford Lecture Series in Mathematics and its Applications, vol. 31. Oxford University Press, Oxford (2006)
98. Parsons, T.D.: Pursuit-evasion in a graph. In: Proceedings Internat. Conf., Western Mich. Univ., Kalamazoo, Mich., 1976, Theory and Applications of Graphs. Lecture Notes in Math., vol. 642, pp. 426–441. Springer, Berlin (1978)
99. Ramachandramurthi, S.: The structure and number of obstructions to treewidth. SIAM J. Discrete Math. 10(1), 146–157 (1997)
100. Reed, B.A.: Finding approximate separators and computing tree width quickly. In: Twenty-Fourth Annual ACM Symposium on Theory of Computing (STOC 1992), pp. 221–228. ACM Press (1992)
101. Reed, B.A., Wood, D.R.: Polynomial treewidth forces a large grid-like-minor. Eur. J. Comb. 33(3), 374–379 (2012)
102. Robertson, N., Seymour, P.D.: Graph minors. I. excluding a forest. J. Combin. Theory, Ser. B 35, 39–61 (1983)
103. Robertson, N., Seymour, P.D.: Graph minors. III. Planar tree-width. J. Combin. Theory, Ser. B 36(1), 49–64 (1984)
104. Robertson, N., Seymour, P.D.: Graph minors. II. algorithmic aspects of tree-width. Journal of Algorithms 7, 309–322 (1986)
105. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. J. Combin. Theory Ser. B 41(1), 92–114 (1986)
106. Robertson, N., Seymour, P.D.: Graph minors. X. Obstructions to tree-decomposition. J. Combin. Theory Ser. B 52(2), 153–190 (1991)
107. Robertson, N., Seymour, P.D.: Graph minors. XXII. Irrelevant vertices in linkage problems (1992) (preprint)
108. Robertson, N., Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Combin. Theory, Ser. B 63(1), 65–110 (1995)
109. Robertson, N., Seymour, P.D.: Graph minors. XVI. Excluding a non-planar graph. J. Combin. Theory Series B 77, 1–27 (1999)
110. Robertson, N., Seymour, P.D.: Graph minors. XX. Wagner's conjecture. J. Combin. Theory Ser. B 92(2), 325–357 (2004)
111. Robertson, N., Seymour, P.D.: Graph minors. XXI. Graphs with unique linkages. J. Combin. Theory Ser. B 99(3), 583–616 (2009)
112. Robertson, N., Seymour, P.D.: Graph minors XXIII. Nash-Williams' immersion conjecture. J. Combin. Theory Ser. B 100(2), 181–205 (2010)
113. Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a planar graph. J. Combin. Theory Ser. B 62(2), 323–348 (1994)
114. Rué, J., Sau, I., Thilikos, D.M.: Dynamic Programming for Graphs on Surfaces. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010, Part I. LNCS, vol. 6198, pp. 372–383. Springer, Heidelberg (2010)
115. Rué, J., Stavropoulos, K.S., Thilikos, D.M.: Outerplanar obstructions for the feedback vertex set. Electronic Notes in Discrete Mathematics 34, 167–171 (2009)

116. Scheffler, P.: A practical linear time algorithm for disjoint paths in graphs with bounded tree-width. Technical Report 396/1994, FU Berlin, Fachbereich 3 Mathematik (1994)
117. Takahashi, A., Ueno, S., Kajitani, Y.: Minimal acyclic forbidden minors for the family of graphs with bounded path-width. Disc. Math. 127(1-3), 293–304 (1994); Graph theory and applications, Hakone (1990)
118. Thilikos, D.M.: Algorithms and obstructions for linear-width and related search parameters. Discrete Applied Mathematics 105, 239–271 (2000)
119. van Leeuwen, J.: Graph algorithms. In: Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A), pp. 525–631. Elsevier Science (1990)
120. van Rooij, J.M.M., Bodlaender, H.L., Rossmanith, P.: Dynamic Programming on Tree Decompositions Using Generalised Fast Subset Convolution. In: Fiat, A., Sanders, P. (eds.) ESA 2009. LNCS, vol. 5757, pp. 566–577. Springer, Heidelberg (2009)
121. Wagner, K.: Über eine eigenschaft der ebenen komplexe. Mathematische Annalen 114, 570–590 (1937), 10.1007/BF01594196
122. Weiss, A.: Girths of bipartite sextet graphs. Combinatorica 4(2-3), 241–245 (1984)