

Parameterized Complexity: Exponential Speed-Up for Planar Graph Problems

Jochen Alber*, Henning Fernau, and Rolf Niedermeier

Universität Tübingen, Wilhelm-Schickard-Institut für Informatik,
Sand 13, 72076 Tübingen, Fed. Rep. of Germany,
{alber,fernau,niedermeier}@informatik.uni-tuebingen.de

Abstract. A parameterized problem is *fixed parameter tractable* if it admits a solving algorithm whose running time on input instance (I, k) is $f(k) \cdot |I|^\alpha$, where f is an arbitrary function depending only on k . Typically, f is some exponential function, e.g., $f(k) = c^k$ for constant c . We describe general techniques to obtain growth of the form $f(k) = c^{\sqrt{k}}$ for a large variety of planar graph problems. The key to this type of algorithm is what we call the “Layerwise Separation Property” of a planar graph problem. Problems having this property include PLANAR VERTEX COVER, PLANAR INDEPENDENT SET, and PLANAR DOMINATING SET.

1 Introduction

While many problems of practical interest tend to be intractable from a standard complexity-theoretic point of view, in many cases such problems have natural “structural” parameters, and practically relevant instances are often associated with “small” values of these parameters. The notion of fixed parameter tractability [10] tries to capture this intuition. This is done by taking into account solving algorithms that are exponential with respect to the parameter, but otherwise have polynomial time complexity. That is, on input instance (I, k) one terms a (parameterized) problem *fixed parameter tractable* if it allows for a solving algorithm running in time $f(k)n^{O(1)}$, where f is an arbitrary function only depending on k and $n = |I|$. The associated complexity class is called FPT. As fixed parameter tractability explicitly allows for exponential time complexity concerning the parameter, the pressing challenge is to keep the related “combinatorial explosion” as small as possible. In this paper, we provide a general framework for NP-hard planar graph problems that allows us to go from typically time $c^k n^{O(1)}$ algorithms to time $c^{\sqrt{k}} n^{O(1)}$ algorithms (subsequently briefly denoted by “ $c^{\sqrt{k}}$ -algorithms”), meaning an exponential speed-up.¹ The main contributions of our work, thus, are

* Supported by the Deutsche Forschungsgemeinschaft (research project PEAL (Parameterized complexity and Exact ALgorithms), NI 369/1-1).

¹ Actually, whenever we can construct a so-called problem kernel of polynomial size in polynomial time (which is often the case for parameterized problems), then we can replace the term $c^{\sqrt{k}} n^{O(1)}$ by $c^{\sqrt{k}} k^{O(1)} + n^{O(1)}$.

- to provide new results and a “structural breakthrough” for the parameterized complexity of a large class of problems,
- to parallel and complement results for the approximability of planar graph problems obtained by Baker [4],
- to methodize and extend previous work on concrete graph problems [1], and
- to systematically compute the bases in the exponential terms.

Fixed parameter tractability. The parameterized tractability approach tries to restrict the seemingly inherent “combinatorial explosion” of NP-hard problems to a “small part” of the input, the parameter. For instance, VERTEX COVER allows for an algorithm with running time $O(kn + 1.3^k)$, where k is the size of the vertex cover to be constructed [8, 14]. One direction in current research is to investigate problems with fixed parameter algorithms of running time $c^k n^{O(1)}$ and to try to get the constant c as small as possible. Getting small constant bases in the exponential factor $f(k)$ is also our concern, but here, we focus on functions f (asymptotically) growing as slowly as possible.

Planar graph problems. Planar graphs build a natural and practically important graph class. Many problems that are NP-complete for general graphs (such as VERTEX COVER and DOMINATING SET) remain so when restricted to planar graphs. Whereas many NP-complete graph problems are hard to approximate in general graphs, Baker, in her well-known work [4], showed that many of them possess a polynomial time approximation scheme for planar graphs. However, the degree of the polynomial grows with the quality of the approximation. Alternatively, finding an “efficient” exact solution in “reasonable exponential time” is an interesting and promising research challenge.

Relations to previous work. In recent work, algorithms were presented that constructively produce a solution for PLANAR DOMINATING SET and related problems in time $c^{\sqrt{k}} n$ [1]. To obtain these results, it was proven that the treewidth of a planar graph with a dominating set of size k is bounded by $O(\sqrt{k})$, and that a corresponding tree decomposition can be found in time $O(\sqrt{k}n)$. Building on that problem-specific work with its rather tailor-made approach for dominating sets, here, we take a much broader perspective. From a practitioner’s point of view, this means that, since the algorithms developed here can be stated in a very general framework, only small parts have to be changed to adapt them to the concrete problem. In this sense, our work differs strongly from research directions where running times of algorithms are improved in a very problem-specific manner (e.g., by extremely sophisticated case-distinctions, as in the case of VERTEX COVER for general graphs). For example, once one can show that a problem has the so-called “Layerwise Separation Property,” one can run a general algorithm which quickly computes a tree decomposition of guaranteed small width (independent of the concrete problem).

Results. We provide a general methodology for the design of $c^{\sqrt{k}}$ -algorithms. A key to this is the notion of select&verify graph problems and the introduction of the Layerwise Separation Property (see Section 3) of such problems in connection with the concept of linear problem kernels (see Subsection 2.1). We show that problems that have the Layerwise Separation Property and admit ei-

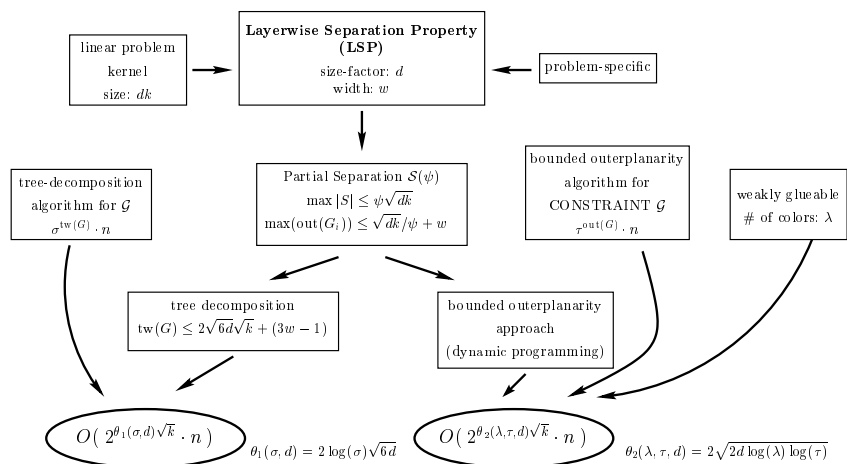


Fig. 1. Roadmap of our methodology for planar graph problems.

ther a tree decomposition based algorithm (cf., e.g., [17]) or admit an algorithm based on bounded outerplanarity (cf. [4]), can be solved in time $c^{\sqrt{k}}n^{O(1)}$. For instance, these include PLANAR VERTEX COVER, PLANAR INDEPENDENT SET, PLANAR DOMINATING SET, or PLANAR EDGE DOMINATION and also variations of these, such as their weighted versions. Moreover, we give explicit formulas to determine the base c of the exponential term with respect to the problem specific parameters. For PLANAR VERTEX COVER, e.g., we obtain a time $O(2^{4\sqrt{3k}}n)$ algorithm. The methods can be generalized in a way that basically all FPT-problems that admit tree-decomposition based algorithms can be attacked with our approach. A library containing implementations of various algorithms sketched in this paper is currently under development. It uses the LEDA package [13] for graph algorithms and the results obtained so far are encouraging.

Review of presented methodology. In a first phase, one separates the graph in a particular way (“layerwise”). The key property of a graph problem which allows such an approach will be the so-called “Layerwise Separation Property.” Corresponding details are presented in Section 3. It will be shown that such a property holds for quite a large class of graph problems. In a second phase, the problem is solved on the layerwisely separated graph. We present two independent ways to achieve this in Section 4; either using the separators to set up a tree decomposition of width $O(\sqrt{k})$ and solving the problem using this tree decomposition, or using a combination of a trivial approach on the separators and some algorithms working on graphs of bounded outerplanarity (see [4]) for the partitioned rest graphs. Figure 1 gives a general overview of our methodology.

Several details and proofs had to be deferred to the full version [2].

2 Basic definitions and preliminaries

We consider undirected graphs $G = (V, E)$, V denoting the vertex set and E denoting the edge set. Sometimes we refer to V by $V(G)$. Let $G[D]$ denote the subgraph induced by a vertex set D . We only consider simple (no double edges) graphs without self-loops. We study *planar* graphs, i.e., graphs that can be drawn in the plane without edge crossings. Let (G, ϕ) denote a *plane* graph, i.e., a planar graph G together with an embedding ϕ . A *face* of a plane graph is any topologically connected region surrounded by edges of the plane graph. The one unbounded face of a plane graph is called the *exterior face*. We study the following “graph numbers”: A *vertex cover* C of a graph G is a set of vertices such that every edge of G has at least one endpoint in C ; the size of a vertex cover set with a minimum number of vertices is denoted by $vc(G)$. An *independent set* of a graph G is a set of pairwise nonadjacent vertices; the size of an independent set with a maximum number of vertices is denoted by $is(G)$. A *dominating set* D of a graph G is a set of vertices such that each of the vertices in G lies in D or has at least one neighbor in D ; the size of a dominating set with a minimum number of vertices is denoted by $ds(G)$. The corresponding problems are (PLANAR) VERTEX COVER, INDEPENDENT SET, and DOMINATING SET.

2.1 Linear problem kernels

Reduction to problem kernel is a core technique for the development of fixed parameter algorithms (see [10]). In a sense, the idea behind is to cut off the “easy parts” of a given problem instance such that only the “hard kernel” of the problem remains, where, then, e.g., exhaustive search can be applied (with reduced costs).

Definition 1. Let \mathcal{L} be a parameterized problem, i.e., \mathcal{L} consists of pairs (I, k) , where problem instance I has a solution of size k (the parameter).² *Reduction to problem kernel*³ then means to replace problem (I, k) by a “reduced” problem (I', k') (which we call the *problem kernel*) such that $k' \leq c \cdot k$ and $|I'| \leq q(k')$ with constant c , polynomial q , and $(I, k) \in \mathcal{L}$ iff $(I', k') \in \mathcal{L}$. Furthermore, we require that the reduction from (I, k) to (I', k') (that we call *kernelization*) is computable in polynomial time $T_K(|I|, k)$.

Usually, having constructed a size $k^{O(1)}$ problem kernel in time $n^{O(1)}$, one can improve the time complexity $f(k)n^{O(1)}$ of a fixed parameter algorithm to $f(k)k^{O(1)} + n^{O(1)}$. Subsequently, our focus is on decreasing $f(k)$, and we do not always refer to this simple fact. Often (cf. the subsequent example VERTEX COVER), the best one can hope for the problem kernel is size linear in k , a so-called *linear problem kernel*. For instance, using a theorem of Nemhauser

² In this paper, we assume the parameter to be a positive integer, although, in general, it might also be from an arbitrary language (e.g., being a subgraph).

³ Here, we give a somewhat “restricted definition” of reduction to problem kernel which, however, applies to all practical cases we know.

and Trotter [15], Chen *et al.* [8] recently observed a problem kernel of size $2k$ for VERTEX COVER on general (not necessarily planar) graphs. According to the current state of knowledge, this is the best one could hope for. As a further example, note that due to the four color theorem for planar graphs and the corresponding algorithm generating a four coloring [16], it is easy to see that PLANAR INDEPENDENT SET has a problem kernel of size $4k$.

Besides the positive effect of reducing the input size significantly, this paper gives further justification, in particular, for the importance of linear problem kernels. The point is that once we have a linear problem kernel, e.g., for PLANAR VERTEX COVER or PLANAR INDEPENDENT SET, it is fairly easy to get $c^{\sqrt{k}}$ -algorithms for these problems based upon the famous planar separator theorem [12]. The constant factor in the problem kernel size directly influences the value of the exponential base and hence, lowering the kernel size means improved efficiency. We will show alternative, more efficient ways (without using the planar separator theorem) of how to make use of linear problem kernels in a generic way in order to obtain $c^{\sqrt{k}}$ -algorithms for planar graph problems.

2.2 Tree decomposition and layer decomposition of graphs

Definition 2. A *tree decomposition* of a graph $G = (V, E)$ is a pair $\langle \{X_i \mid i \in I\}, T \rangle$, where $X_i \subseteq V$ is called a *bag* and T is a tree with the elements of I as nodes, such that the following hold:

1. $\bigcup_{i \in I} X_i = V$;
2. for every edge $\{u, v\} \in E$, there is an $i \in I$ such that $\{u, v\} \subseteq X_i$;
3. for all $i, j, k \in I$, if j lies on the path between i and k in T , then $X_i \cap X_k \subseteq X_j$.

The *width* of $\langle \{X_i \mid i \in I\}, T \rangle$ is $\max\{|X_i| \mid i \in I\} - 1$. The *treewidth* $\text{tw}(G)$ of G is the minimum ℓ such that G has a tree decomposition of width ℓ .

Details on tree decompositions can be found in [5, 6, 11]. Let $G = (V, E)$ be a planar graph. The vertices of G can be decomposed according to the level of the “layer” in which they appear in an embedding ϕ , see [1, 4].

Definition 3. Let $(G = (V, E), \phi)$ be a plane graph.

- a) The *layer decomposition* of (G, ϕ) is a disjoint partition of the vertex set V into sets L_1, \dots, L_r , which are recursively defined as follows:
 - L_1 is the set of vertices on the exterior face of G , and
 - L_i is the set of vertices on the exterior face of $G[V - \bigcup_{j=1}^{i-1} L_j]$ for $i = 2, \dots, r$.

We will denote the layer decomposition of (G, ϕ) by $\mathcal{L}(G, \phi) := (L_1, \dots, L_r)$.

- b) The set L_i is called the *i th layer* of (G, ϕ) .
- c) The (uniquely defined) number r of different layers is called the *outerplanarity* of (G, ϕ) , denoted by $\text{out}(G, \phi) := r$.
- d) We define $\text{out}(G)$ to be the smallest outerplanarity possible among all plane embeddings, i.e., minimizing over all plane embeddings ϕ of G we set

$$\text{out}(G) := \min_{\phi} \text{out}(G, \phi).$$

Proposition 4 ([1]). *Let $(G = (V, E), \phi)$ be a plane graph. The layer decomposition $\mathcal{L}(G, \phi) = (L_1, \dots, L_r)$ can be computed in time $O(|V|)$.*

2.3 Algorithms based on separators in graphs

One of the most useful algorithmic techniques for solving computational problems is divide-and-conquer. To apply this technique to planar graphs, we need graph separators and related notions.

Graph separators and select&verify problems. Graph separators are defined as follows. Let $G = (V, E)$ be an undirected graph. A *separator* $S \subseteq V$ of G partitions V into two sets A and B such that $A \cup B \cup S = V$ with $A \cap B = A \cap S = B \cap S = \emptyset$ and no edge joins vertices in A and B . In general, of course, A , B and S will be non-empty.

Definition 5. A set \mathcal{G} of tuples (G, k) , G an undirected graph with vertex set $V = \{v_1, \dots, v_n\}$ and k a positive real number, is called a *select&verify graph problem* if there exists a pair (P, opt) with $\text{opt} \in \{\min, \max\}$, such that P is a function that assigns to an undirected graph G (with n vertices) a polynomial time computable function $P_G : \{0, 1\}^n \rightarrow \mathbb{R}_+ \cup \{\pm\infty\}$, such that

$$(G, k) \in \mathcal{G} \quad \Leftrightarrow \quad \begin{cases} \text{opt}_{\mathbf{x} \in \{0,1\}^n} P_G(\mathbf{x}) \leq k & \text{if } \text{opt} = \min \\ \text{opt}_{\mathbf{x} \in \{0,1\}^n} P_G(\mathbf{x}) \geq k & \text{if } \text{opt} = \max. \end{cases}$$

It is an easy observation that every select&verify graph problem that additionally admits a linear problem kernel of size dk is solvable in time $O(2^{dk}k + T_K(n, k))$.

VERTEX COVER is an easy example for a select&verify graph problem. Here, for $G = (V, E)$, one may use (with the convention $\infty \cdot 0 = 0$)

$$P_G(\mathbf{x}) = \sum_{i=1}^n x_i + \sum_{\{v_i, v_j\} \in E} \infty \cdot (1 - x_i)(1 - x_j).$$

Algorithms based on separator theorems. Lipton and Tarjan [12] have used their famous separator theorem in order to design algorithms with a running time of $O(c\sqrt{n})$ for certain select&verify planar graph problems. This naturally implies that, in the case of parameterized planar graph problems for which a linear kernel is known, algorithms with running time $O(c'\sqrt{k} + T_K(n, k))$ can be derived. As worked out in [3], a straightforward application yields very bad constants, even when dealing with improved versions of the planar separator theorem (see [9]); for instance, $c' = 2^{15.1823} \approx 40000$ for PLANAR VERTEX COVER. We will see algorithms with much better constants in this paper. In addition, the advantages of the approach pursued in this paper also lie in weaker assumptions. In some cases, we may drop requirements such as linear problem kernels by replacing it with the so-called “Layerwise Separation Property,” a seemingly less restrictive demand.

3 Phase 1: Layerwise separation

We will exploit the layer-structure of a plane graph in order to gain a “nice” separation of the graph. It is important that a “yes”-instance (G, k) (where G is a plane graph) of the graph problem \mathcal{G} admits a so-called “layerwise separation” of small size. By this, we mean, roughly speaking, a separation of the plane graph G (i.e., a collection of separators for G), such that each separator is contained in the union of constantly many subsequent layers (see conditions 1 and 2 of the following definition). For (fixed parameter) algorithmic purposes, it will be important that the corresponding separators are “small” (see condition 3 below).

Definition 6. Let $(G = (V, E), \phi)$ be a plane graph of outerplanarity $r := \text{out}(G, \phi)$ and let $\mathcal{L}(G, \phi) = (L_1, \dots, L_r)$ be its layer decomposition. A *layerwise separation of width w and size s* of (G, ϕ) is a sequence (S_1, \dots, S_r) of subsets of V , with the properties that:

1. $S_i \subseteq \bigcup_{j=i}^{i+(w-1)} L_j$, 2. S_i separates layers L_{i-1} and L_{i+w} , and 3. $\sum_{j=1}^r |S_j| \leq s$.

Definition 7. A parameterized problem \mathcal{G} for planar graphs is said to have the *Layerwise Separation Property* (abbreviated by: LSP) of width w and size-factor d if for each $(G, k) \in \mathcal{G}$ and every planar embedding ϕ of G , the plane graph (G, ϕ) admits a layerwise separation of width w and size dk .

3.1 How can layerwise separations be obtained?

The Layerwise Separation Property can be shown directly for many parameterized graph problems. As an example, consider PLANAR VERTEX COVER. Here, we get constants $w = 2$ and $d = 2$. In fact, for $(G, k) \in \text{VERTEX COVER}$ (where (G, ϕ) is a plane graph) with a “witnessing” vertex cover V' of size k , the sets $S_i := (L_i \cup L_{i+1}) \cap V'$ form a layerwise separation, given the layer decomposition $\mathcal{L}(G, \phi) = (L_1, \dots, L_r)$. In [1], the non-trivial fact is proven that for PLANAR DOMINATING SET, the LSP holds with constants $w = 3$ and $d = 51$.

Lemma 8. *Let \mathcal{G} be a parameterized problem for planar graphs that admits a problem kernel of size dk . Then, the parameterized problem \mathcal{G}' where each instance is replaced by its problem kernel has the LSP of width 1 and size-factor d .*

With Lemma 8 and the size $2k$ problem kernel for VERTEX COVER (see Subsection 2.1), we derive, for example, that PLANAR VERTEX COVER has the LSP of width 1 and size-factor 2 (which is even better than what was shown above). Using the $4k$ problem kernel for PLANAR INDEPENDENT SET, we see that this problem has the LSP of width 1 and size-factor 4 on the set of reduced instances.

3.2 What are layerwise separations good for?

The idea of the following is that, from a layerwise separation of small size (say bounded by $O(k)$), we are able to choose a set of separators such that their size is bounded by $O(\sqrt{k})$ and—at the same time—the subgraphs into which these separators cut the original graph have outerplanarity bounded by $O(\sqrt{k})$.

Definition 9. Let $(G = (V, E), \phi)$ be a plane graph with layer decomposition $\mathcal{L}(G, \phi) = (L_1, \dots, L_r)$. A *partial layerwise separation of width w* is a sequence $\mathcal{S} = (S_1, \dots, S_q)$ such that there exist $i_0 = 1 \leq i_1 < \dots < i_q \leq r = i_{q+1}$ such that for $i = 1, \dots, q$:⁴

1. $S_j \subseteq \bigcup_{\ell=i_j}^{i_{j+1}+(w-1)} L_\ell$,
2. $i_j + w \leq i_{j+1}$ (so the sets in \mathcal{S} are pairwise disjoint), and
3. S_j separates layers $L_{i_{j-1}}$ and L_{i_j+w} .

The sequence $\mathcal{C}_\mathcal{S} = (G_0, \dots, G_q)$ with

$$G_j := G\left[\left(\bigcup_{\ell=i_j}^{i_{j+1}+(w-1)} L_\ell\right) - (S_j \cup S_{j+1})\right], \quad j = 0, \dots, q$$

is called the *sequence of graph chunks* obtained by \mathcal{S} .

Theorem 10. Let $(G = (V, E), \phi)$ be a plane graph that admits a layerwise separation of width w and size dk . Then, for every $\psi \in \mathbb{R}_+$, there exists a partial layerwise separation $\mathcal{S}(\psi)$ of width w such that

1. $\max_{S \in \mathcal{S}(\psi)} |S| \leq \psi\sqrt{dk}$ and
2. $\text{out}(H) \leq \frac{\sqrt{dk}}{\psi} + w$ for each graph chunk H in $\mathcal{C}_{\mathcal{S}(\psi)}$.

Moreover, there is an algorithm with running time $O(\sqrt{kn})$ which, for given ψ , recognizes whether (G, ϕ) admits a layerwise separation of width w and size dk and, if so, computes $\mathcal{S}(\psi)$.

Proof. (Sketch) For $m = 1, \dots, w$, consider the integer sequences $I_m = (m + jw)_{j=0}^{\lfloor r/w \rfloor - 1}$ and the corresponding sequences of separators $\mathcal{S}_m = (S_i)_{i \in I_m}$. Note that each \mathcal{S}_m is a sequence of pairwise disjoint separators. Since (S_1, \dots, S_r) is a layerwise separation of size dk , this implies that there exists a $1 \leq m' \leq w$ with $\sum_{i \in I_{m'}} |S_i| \leq \frac{dk}{w}$ (*).

For a given ψ , let $s := \psi\sqrt{dk}$. Define $\mathcal{S}(\psi)$ to be the subsequence of $\mathcal{S}_{m'}$ such that $|S| \leq s$ for all $S \in \mathcal{S}(\psi)$, and $|S| > s$ for all $S \in \mathcal{S}_{m'} - \mathcal{S}(\psi)$. This yields condition 1. As to condition 2, suppose that $\mathcal{S}(\psi) = (S_{i_1}, \dots, S_{i_q})$. The number of separators in $\mathcal{S}_{m'}$ that appear between S_{i_j} and $S_{i_{j+1}}$ is $(i_{j+1} - i_j)/w$. Since all of these separators have size $\geq s$, their number has to be bounded by dk/ws , see (*). Therefore, $i_{j+1} - i_j \leq \sqrt{dk}/\psi$ for all $j = 1, \dots, q - 1$. Hence, the chunks $G\left[\left(\bigcup_{\ell=i_j}^{i_{j+1}+(w-1)} L_\ell\right) - (S_{i_j} \cup S_{i_{j+1}})\right]$ have outerplanarity at most $\sqrt{dk}/\psi + w$.

The proof can be turned into a constructive algorithm. This is outlined in the full version [2]. □

4 Phase 2: Algorithms on layerwisely separated graphs

After Phase 1, we are left with a set of disjoint (layerwise) separators of size $O(\sqrt{k})$ separating the graph in components, each of which having outerplanarity bounded by $O(\sqrt{k})$.

⁴ By default, we set $S_i := \emptyset$ for $i < 1$ and $i > q$.

4.1 Using tree decompositions

We will show how the existence of a layerwise separation of small size helps to constructively obtain a tree decomposition of small width. The following result can be found in [6, Theorem 83] and [1, Theorem 12].

Proposition 11. *For a plane graph (G, ϕ) , we have $\text{tw}(G) \leq 3 \cdot \text{out}(G) - 1$. Such a tree decomposition can be found in $O(\text{out}(G) \cdot n)$ time.*

Theorem 12. *Let (G, ϕ) be a plane graph that admits a layerwise separation of width w and size dk . Then, we have $\text{tw}(G) \leq 2\sqrt{6dk} + (3w - 1)$. Such a tree decomposition can be computed in time $O(k^{3/2}n)$.*

Proof. (Sketch) By Theorem 10, for each $\psi \in \mathbb{R}_+$, there exists a partial layerwise separation $\mathcal{S}(\psi) = (S_1, \dots, S_q)$ of width w with corresponding graph chunks $\mathcal{C}_{\mathcal{S}(\psi)} = (G_0, \dots, G_q)$, such that $\max_{S \in \mathcal{S}(\psi)} |S| \leq \psi\sqrt{dk}$ and $\text{out}(G_i) \leq \sqrt{dk}/\psi + w$ for $i = 0, \dots, q$. The algorithm that constructs a tree decomposition \mathfrak{X}_ψ is:

1. Construct a tree decomposition \mathcal{X}_i of width at most $3 \text{out}(G_i) - 1$ for each of the graphs G_i (using the algorithm from Proposition 11).
2. Add S_i and S_{i+1} to every bag in \mathcal{X}_i ($i = 0, \dots, q$).
3. Let T_i be the tree of \mathcal{X}_i . Then, successively add an arbitrary connection between the trees T_i and T_{i+1} in order to obtain a tree T .

The tree T , together with the constructed bags, gives a tree decomposition of G , see [1, Prop. 4]. Its width $\text{tw}(\mathfrak{X}_\psi)$ is upperbounded by $(2\psi + 3/\psi)\sqrt{dk} + (3w - 1)$, which is minimal if $\psi = \sqrt{3/2}$. Therefore, $\text{tw}(\mathfrak{X}_\psi) \leq 2\sqrt{6dk} + (3w - 1)$. \square

For example, Theorem 12 and previous observations imply $\text{tw}(G) \leq 4\sqrt{3vc(G)} + 5$ and $\text{tw}(G) \leq 6\sqrt{34ds(G)} + 8$ for planar graphs G . Note that for general graphs, no relation of the form $\text{tw}(G) \leq f(ds(G))$ (for any function f) holds. For VERTEX COVER, only the linear relation $\text{tw}(G) \leq vc(G)$ can be shown easily.

In addition, Theorem 12 yields a $c\sqrt{k}$ -algorithm for certain graph problems.

Theorem 13. *Let \mathcal{G} be a parameterized problem for planar graphs. Suppose that \mathcal{G} has the LSP of width w and size-factor d and that there exists a time $\sigma^\ell n$ algorithm that decides $(G, k) \in \mathcal{G}$, if G is given together with a tree decomposition of width ℓ .*

Then, there is an algorithm to decide $(G, k) \in \mathcal{G}$ in time $O(\sigma^{3w-1} \cdot 2^{\theta_1(\sigma, d)\sqrt{k}} n)$, where $\theta_1(\sigma, d) = 2 \log(\sigma)\sqrt{6d}$.

Proof. In time $O(\sqrt{k}n)$ (see Theorem 10), we can check whether an instance (G, k) admits a layerwise separation of width w and size dk . If so, the algorithm of Theorem 12 computes a tree decomposition of width at most $2\sqrt{6dk} + (3w - 1)$, and we can decide $(G, k) \in \mathcal{G}$ by using the given tree decomposition algorithm in time $O(\sigma^{2\sqrt{6dk} + (3w-1)} n)$. If (G, k) does not admit such a layerwise separation, we know that $(G, k) \notin \mathcal{G}$, by definition of LSP. \square

Going back to our running examples, it is well-known that PLANAR VERTEX COVER and PLANAR INDEPENDENT SET admit such a tree decomposition based algorithm for $\sigma = 2$. For PLANAR VERTEX COVER, we have seen that the LSP of width 1 and size-factor $d = 2$ holds. Hence, Theorem 13 guarantees an $O(2^{4\sqrt{3k}}n)$ algorithm for this problem. For PLANAR INDEPENDENT SET, we have a linear problem kernel of size $4k$, hence, the LSP of width 1 and size-factor $d = 4$ holds, which yields an $O(2^{4\sqrt{6k}}n)$ algorithm.

4.2 Using bounded outerplanarity

We now turn our attention to select&verify problems subject to the assumption that a solving algorithm of linear running time on the class of graphs of bounded outerplanarity exists. This issue was addressed in [4]; a variety of examples can be found therein. We examine how, in this context, the notions of select&verify problems and the LSP will lead to $c^{\sqrt{k}}$ -algorithms.

Due to the lack of space, we only give an intuitive explanation of the notions “weak glueability” and “CONSTRAINT \mathcal{G} ” associated to a select&verify problem \mathcal{G} which appear in the formulation of the following results. For a more detailed definition we refer to the long version [2] or to [3]. A problem \mathcal{G} is *weakly glueable with λ colors* if a solution of \mathcal{G} on an instance G can be obtained by “merging” solutions of CONSTRAINT \mathcal{G} with $G[AUS]$ and $G[BUS]$, where S separates G into two parts A and B . Here, CONSTRAINT \mathcal{G} is a variant of \mathcal{G} , in which it is already fixed which vertices of S belong to an admissible solution. The number λ , in some sense, measures the complexity of the merging step. For example, PLANAR VERTEX COVER, and PLANAR INDEPENDENT SET are weakly glueable with $\lambda = 2$ colors and, PLANAR DOMINATING SET is weakly glueable with “essentially” $\lambda = 3$ colors.

Similar to Theorem 13, we construct a partial layerwise separation $\mathcal{S}(\psi)$ with optimally adapted trade-off parameter ψ to enable an efficient dynamic programming algorithm. We omit the proof of the following theorem (see [2] for details).

Theorem 14. *Let \mathcal{G} be a select&verify problem for planar graphs. Suppose that \mathcal{G} has the LSP of width w and size-factor d , that \mathcal{G} is weakly glueable with λ colors, and that there exists an algorithm that solves the problem CONSTRAINT \mathcal{G} for a given graph G in time $\tau^{\text{out}(G)}n$.*

Then, there is an algorithm to decide $(G, k) \in \mathcal{G}$ in time $O(\tau^w \cdot 2^{\theta_2(\lambda, \tau, d)\sqrt{k}}n)$, where $\theta_2(\lambda, \tau, d) = 2\sqrt{2d \log(\lambda) \log(\tau)}$.

It remains to say for which problems there exists a solving algorithm of the problem CONSTRAINT \mathcal{G} for a given graph G in time $\tau^{\text{out}(G)}n$. For PLANAR VERTEX COVER, we have $d = 2$, $w = 1$ and $\tau = 8$ (see the result of Baker [4] which can be adapted to the constraint case fairly easily) and, hence, the approach in Theorem 14 yields an $O(2^{4\sqrt{3k}}n)$ time algorithm.

As an alternative to Baker, we again may use tree decomposition based approaches: Let \mathcal{G} be a parameterized problem for planar graphs. Suppose that

there exists a time $\sigma^\ell n$ algorithm that solves CONSTRAINT \mathcal{G} , when G is given together with a tree decomposition of width ℓ . Then, due to Proposition 11, there is an algorithm that solves CONSTRAINT \mathcal{G} in time $\tau^{\text{out}(G)} n$ for $\tau = \sigma^3$.

The following easy corollary helps comparing the approach from Subsection 4.1 (i.e., Theorem 13) with the approach in this subsection (i.e., Theorem 14).

Corollary 15. *Let \mathcal{G} be a select&verify problem for planar graphs. Suppose that \mathcal{G} has the LSP of width w and size-factor d , that \mathcal{G} is weakly glueable with λ colors, and that there exists a time $\sigma^\ell n$ algorithm that solves CONSTRAINT \mathcal{G} for a graph G , if G is given together with a tree decomposition of width ℓ .*

Then, there is an algorithm to decide $(G, k) \in \mathcal{G}$ in time $O(\sigma^{3w} \cdot 2^{\theta_3(\lambda, \sigma, d)\sqrt{k}} n)$, where $\theta_3(\lambda, \sigma, d) = 2\sqrt{6d \log(\lambda) \log(\sigma)}$.

The exponential factor of the algorithm in Corollary 15, i.e., $\theta_3(\lambda, \sigma, d)$, is related to the corresponding exponent of Theorem 13, i.e., $\theta_1(\sigma, d)$, in the following way: $\sqrt{\log \lambda} \cdot \theta_1(\sigma, d) = \sqrt{\log \sigma} \cdot \theta_3(\lambda, \sigma, d)$. From this, we derive that, if $\lambda > \sigma$, the algorithm in Theorem 13 outperforms the one of Corollary 15, whereas, if $\lambda < \sigma$, the situation is vice versa. However, in order to apply Corollary 15, we need the three extra assumptions that we have a select&verify problem which is weakly glueable and that we can deal with the problem CONSTRAINT \mathcal{G} in the treewidth algorithm.

5 Conclusion

To some extent, this paper can be seen as the “parameterized complexity counterpart” to what was developed by Baker [4] in the context of approximation algorithms. We describe two main ways (namely linear problem kernels and problem-specific approaches) to achieve the novel concept of Layerwise Separation Property, from which again, two approaches (tree decomposition and bounded outerplanarity) lead to $c^{\sqrt{k}}$ -algorithms for planar graph problems (see Figure 1 for an overview). A slight modification of our presented techniques can be used to extend our results to parameterized problems that admit a problem kernel of size $p(k)$ (not necessarily linear!). In this case, the running time can be sped up from $2^{O(p(k))} n^{O(1)}$ to $2^{O(\sqrt{p(k)})} n^{O(1)}$ (see [2] for details). Basically all FPT-problems that admit treewidth based algorithms can be handled by our methods (see [17]).

Future research topics raised by our work include to further improve the (“exponential”) constants, e.g., by a further refined and more sophisticated “layer decomposition tree”; to investigate and extend the availability of linear problem kernels for all kinds of planar graph problems; to provide implementations of our approach accompanied by sound experimental studies, thus taking into account that all our analysis is worst case and often overly pessimistic. Finally, a more general question is whether there are other “problem classes” that allow for $c^{\sqrt{k}}$ fixed parameter algorithms. Cai and Juedes [7], however, very recently showed

the surprising result that for a list of parameterized problems (e.g., VERTEX COVER on general graphs) $c^{o(k)}$ -algorithms are impossible unless $\text{FPT} = W[1]$.

Acknowledgements. We'd like to mention that parts of this work were discussed at the first international Workshop on Parameterized Complexity (organized by Mike Fellows and Venkatesh Raman) in Chennai, India, December 7–9, 2000.

References

1. J. Alber, H. Bodlaender, H. Fernau, and R. Niedermeier. Fixed parameter algorithms for planar dominating set and related problems. In *Proc. 7th SWAT*, vol. 1851 of *LNCS*, Springer, pp. 97–110, 2000. Full version available as Technical Report UU-CS-2000-28, Utrecht University, 2000.
2. J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. Technical Report TR01-023, ECCO Reports, Trier, March 2001. Available through <http://www.eccc.uni-trier.de/eccc/>.
3. J. Alber, H. Fernau, and R. Niedermeier. Graph separators: a parameterized view. To appear in *Proc. 7th COCOON*, 2001. Full version available as Technical Report WSI-2001-8, Universität Tübingen (Germany), Wilhelm-Schickard-Institut für Informatik, March 2001.
4. B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.
5. H. L. Bodlaender. Treewidth: Algorithmic techniques and results. In *Proc. 22nd MFCS*, vol. 1295 of *LNCS*, Springer, pp. 19–36, 1997.
6. H. L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comp. Sci.*, 209:1–45, 1998.
7. L. Cai and D. Juedes. Subexponential parameterized algorithms collapse the W-hierarchy. In *Proc. 28th ICALP*, 2001.
8. J. Chen, I. Kanj, and W. Jia. Vertex cover: Further observations and further improvements. In *Proc. 25th WG*, vol. 1665 of *LNCS*, Springer, pp. 313–324, 1999.
9. H. N. Djidjev and S. Venkatesan. Reduced constants for simple cycle graph separation. *Acta Informatica*, 34:231–243, 1997.
10. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
11. T. Kloks. *Treewidth: Computations and Approximations*, vol. 842 of *LNCS*, Springer, 1994.
12. R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comp.*, 9(3):615–627, 1980.
13. K. Mehlhorn and S. Näher. *LEDA: A Platform of Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, England, 1999.
14. R. Niedermeier and P. Rossmanith. Upper Bounds for Vertex Cover further improved. In *Proc. 16th STACS*, vol. 1563 of *LNCS*, Springer, pp. 561–570, 1999.
15. G. L. Nemhauser and J. L. E. Trotter. Vertex packing: structural properties and algorithms. *Math. Progr.*, 8:232–248, 1975.
16. N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas. Efficiently four-coloring planar graphs. In *Proc. 28th STOC*, ACM Press, pp. 571–575, 1996.
17. J. A. Telle and A. Proskurowski. Practical algorithms on partial k -trees with an application to domination-like problems. In *Proc. 3rd WADS*, vol. 709 of *LNCS*, Springer, pp. 610–621, 1993.