LCNS, Vol 2108, pp. 318–327, 2001

Graph Separators: A Parameterized View*

Jochen Alber**, Henning Fernau, and Rolf Niedermeier

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

Abstract. Graph separation is a well-known tool to make (hard) graph problems accessible for a divide and conquer approach. We show how to use graph separator theorems in order to develop fixed parameter algorithms for many well-known NP-hard (planar) graph problems. We coin the key notion of glueable select&verify graph problems and derive from that a prospective way to easily check whether a planar graph problem will allow for a fixed parameter divide and conquer algorithm of running time $c^{\sqrt{k}} \cdot n^{O(1)}$ for a constant c.

1 Introduction

Algorithm designers are often faced with problems which, when viewed from classical computational complexity theory, are "intractable." More formally speaking, these problems can be shown to be *NP*-hard. In many applications, however, a certain part (called the *parameter*) of the whole problem can be identified which tends to be of small size k when compared with the size n of the whole problem instance. This leads to the study of parameterized complexity [7].

Fixed parameter tractability. Formally, one terms a (parameterized) problem *fixed parameter tractable* if it allows for a solving algorithm running in time $f(k)n^{O(1)}$ on input instance (I, k), where $n = |I|$ and f is an arbitrary function only depending on k. We will also term such algorithms "$f(k)$-algorithms" for brevity, focusing on the exponential part of the running time bound. The associated complexity class is called FPT. Of course, designing fixed parameter algorithms with a "small" function f is desirable. To our knowledge, so far, only one non-trivial fixed parameter tractability result where the corresponding function f is sublinear in the exponent, namely $f(k) = c^{\sqrt{k}}$ [1], is known: PLANAR DOMINATING SET. Similar results hold for closely related problems such as FACE COVER, PLANAR INDEPENDENT DOMINATING SET, PLANAR WEIGHTED DOMINATING SET, etc. [1]. In the companion paper [2], we proved similar results for a much broader class of planar graph problems, presenting a general methodology based on concepts such as tree decompositions and bounded outerplanarity.

* Full version: Technical Report No. WSI–2001–8, Universität Tübingen (Fed. Rep. of Germany), Wilhelm-Schickard-Institut für Informatik.

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

Here, we will also discuss a rather general approach for obtaining parameterized graph algorithms running in time[1] $O(c^{e(k)} \cdot q(n))$ for sublinear functions e, i.e., $e(k) \in o(k)$. More precisely, we investigate the use of (planar) separator theorems in this context, yielding an alternative and conceptually rather different framework in comparison with [2].

General outline. It is a crucial goal throughout the paper not to narrowly stick to problem-specific approaches, but to try to widen the techniques as far as possible. More specifically, we show how to use separator theorems for different graph classes, such as, e.g., the well-known planar separator theorem due to Lipton and Tarjan [8], in combination with known algorithms for obtaining (linear size) problem kernels in order to obtain fixed parameter divide and conquer algorithms. Special care is taken for the dependency of the running time on the "graph separator parameters" (and how they influence the recurrences in the running time analysis). We consider a broad class of problems that can be attacked by this approach, namely, in principle, all so-called glueable select&verify problems such as, e.g., PLANAR DOMINATING SET.[2] Also, we exhibit the influence on the running time analysis of so-called cycle separators. Although the constants achieved in our setting so far seem to be too large in order to yield practical algorithms, our approach provides a general, sound, mathematical formalization of a rich class of problems that allow for divide and conquer fixed parameter algorithms. Our methodology seems to leave much room for improvement in many directions. For instance, we introduce the novel concept of problem cores that can replace problem kernels in our setting. Finally, we give a push to the study of subclasses of the parameterized complexity class FPT. In this sense, our work also might serve as a starting point for more algorithmic (also concerning graph theory with respect to separator theorems), as well as more structural complexity-theoretic lines of future research in parameterized complexity.

Due to the lack of space, several details are deferred to the long version.

2 Basic definitions and preliminaries

We consider undirected graphs $G = (V, E)$, V denoting the vertex set and E denoting the edge set. We only consider simple graphs (i.e., with no double edges) without self-loops. Sometimes, we refer to V by $V(G)$ in order to emphasize that V is the vertex set of graph G; by $N(v)$ we refer to the set of vertices adjacent to v. $G[D]$ denotes the subgraph induced by a vertex set D. For graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, by $G_1 \cap G_2$, we denote the graph $(V_1 \cap V_2, E_1 \cap E_2)$. $G' = (V', E')$ is a subgraph of $G = (V, E)$, denoted by $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. In this paper, we only consider graph classes, denoted by $\mathbb{G}$, that are closed under taking subgraphs. The most important among these graph classes

[1] Here and in the following, n is the number of vertices of the input graph and k is the parameter of the considered graph problem.

[2] Lipton and Tarjan [9] only describe a solution for the structurally much simpler PLANAR INDEPENDENT SET in detail.

is that of *planar* graphs, i.e., graphs that have a drawing in the plane without edge crossings. Among others, we study the following vertex subsets of a graph $G = (V, E)$: A *vertex cover* $C \subseteq V$ satisfies that every edge of G has at least one endpoint in C. An *independent set* is a set of pairwise nonadjacent vertices. A *dominating set* $D \subseteq V$ obeys that each of the rest of the vertices in G has at least one neighbor in D. The corresponding problems are denoted by (PLANAR) VERTEX COVER, INDEPENDENT SET, and DOMINATING SET.

Further conventions: we write $A+B$ to denote the disjoint union of sets A and B. We let $0 \cdot (\pm\infty) = 0$.

Linear problem kernels. Let $\mathcal{L}$ be a parameterized problem, i.e., $\mathcal{L}$ is a subset of $\Sigma^* \times \mathbb{N}$.[3] *Reduction to problem kernel*, then, means to replace instance $(I, k) \in \Sigma^* \times \mathbb{N}$ by a "reduced" instance $(I', k') \in \Sigma^* \times \mathbb{N}$ (which we call *problem kernel*) such that $k' \leq c \cdot k, |I'| \leq p(k)$ with constant c, some function p only depending on k, and $(I, k) \in \mathcal{L}$ iff $(I', k') \in \mathcal{L}$. Furthermore, we require that the reduction from (I, k) to (I', k') is computable in polynomial time $T_K(|I|, k)$.

Often, the best one can hope for is that the problem kernel is size linear in k, a so-called *linear problem kernel*. For instance, using a theorem of Nemhauser and Trotter, Chen *et al.* [4] observed a problem kernel of size $2k$ for VERTEX COVER on general (not necessarily planar) graphs. Furthermore, PLANAR INDEPENDENT SET has a problem kernel of size $4k$ due to the four color theorem. Once having a linear size problem kernel, it is fairly easy to use our framework to get $c^{\sqrt{k}}$-algorithms for these problems based upon the famous planar separator theorem [8, 9]. The constant factor in the problem kernel size directly influences the value of the exponential base. Hence, lowering the kernel size is a crucial goal.

Classical separator theorems. Let $G = (V, E)$ be an undirected graph. A *separator* $S \subseteq V$ of G partitions V into two *parts* $A_1 \subseteq V$ and $A_2 \subseteq V$ such that $A_1 + S + A_2 = V$, and no edge joins vertices in A_1 and A_2; (A_1, S, A_2) is called a *separation* of G. When we restrict our attention to *planar graphs*, S is called *cycle separator* if it forms a cycle in some triangulation of G. According to Lipton and Tarjan [8], an *$f(\cdot)$-separator theorem (with constants $\alpha < 1$, $\beta > 0$)* for a class $\mathbb{G}$ of graphs which is closed under taking subgraphs is a theorem of the following form: If G is any n-vertex graph in $\mathbb{G}$, then there is a separation (A_1, S, A_2) of G such that neither A_1 nor A_2 contains more than αn vertices, and S contains no more than $\beta f(n)$ vertices.

Stated in this framework, the planar separator theorem due to Lipton and Tarjan [8] is a $\sqrt{\cdot}$-separator theorem with constants $\alpha = 2/3$ and $\beta = 2\sqrt{2}$. The current record for $\alpha = 2/3$ is $\beta \approx 1.97$ [6]. Similar $\sqrt{\cdot}$-separator theorems are also known for other graph classes, e.g., for the class of graphs of bounded genus, see [5]. It is also possible to incorporate *weights* in most separator theorems. We refer to them as $f(\cdot)$-separator theorems for weighted graphs.

[3] 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 Glueable graph problems

Select&verify graph problems. A set $\mathcal{G}$ of tuples (G,k), G an undirected graph with vertex set $V = \{v_1, \dots, v_n\}$ and k a nonnegative real number, is called a *select&verify (graph) problem* if there exists a pair $(P_\cdot, \mathrm{opt})$ with $\mathrm{opt} \in \{\min, \max\}$, such that $P_\cdot$ is a function that assigns to G a polynomial time computable function of the form $P_G = P_G^{\mathrm{sel}} + P_G^{\mathrm{ver}}$, where $P_G^{\mathrm{sel}} : \{0,1\}^n \to \mathbb{R}_+$, $P_G^{\mathrm{ver}} : \{0,1\}^n \to \{0, \pm\infty\}$, and

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

For $\boldsymbol{x} = (x_1, \dots, x_n) \in \{0,1\}^n$ with $P_G(\boldsymbol{x}) \le k$ if $\mathrm{opt} = \min$ and with $P_G(\boldsymbol{x}) \ge k$ if $\mathrm{opt} = \max$, the vertex set *selected* by $\boldsymbol{x}$ and *verified* by P_G is $\{v_i \in V \mid x_i = 1, 1 \le i \le n\}$. A vector $\boldsymbol{x}$ is called *admissible* if $P_G^{\mathrm{ver}}(\boldsymbol{x}) = 0$.

The intuition behind the term $P_\cdot = P_\cdot^{\mathrm{sel}} + P_\cdot^{\mathrm{ver}}$ is that the "selecting function" $P_\cdot^{\mathrm{sel}}$ counts the size of the selected set of vertices and the "verifying function" $P_\cdot^{\mathrm{ver}}$ verifies whether this choice of vertices is an admissible solution. Every select&verify graph problem that additionally admits a problem kernel of size $p(k)$ is solvable in time $O(2^{p(k)}p(k) + T_K(n,k))$.

We now give two examples for select&verify problems by specifying the function $P_G = P_G^{\mathrm{sel}} + P_G^{\mathrm{ver}}$. In both cases the "selecting function" for a graph $G = (V,E)$ will be $P_G^{\mathrm{sel}} = \sum_{v_i \in V} x_i$. Firstly, in the case of Vertex cover, we have $\mathrm{opt} = \min$ and choose $P_G^{\mathrm{ver}}(\boldsymbol{x}) = \sum_{\{v_i,v_j\}\in E} \infty \cdot (1-x_i)(1-x_j)$. Thus, $P_G(\boldsymbol{x}) \le k$ guarantees a size at most k vertex cover set. Secondly, for dominating set, we have $P_G^{\mathrm{ver}}(\boldsymbol{x}) = \sum_{v_i\in V}(\infty \cdot (1-x_i) \cdot \prod_{\{v_i,v_j\}\in E}(1-x_j))$.

We will also need a notion of select&verify problems where the "selecting function" and the "verifying function" operate on a subgraph of the given graph: Let $P_\cdot = P_\cdot^{\mathrm{sel}} + P_\cdot^{\mathrm{ver}}$ be the function of a select&verify problem. For an n-vertex graph G and subgraphs $G^{\mathrm{ver}} = (V^{\mathrm{ver}}, E^{\mathrm{ver}})$, $G^{\mathrm{sel}} = (V^{\mathrm{sel}}, E^{\mathrm{sel}}) \subseteq G$, we let

$$P_{G^{\mathrm{ver}}}(\boldsymbol{x} \mid G^{\mathrm{sel}}) := P_{G^{\mathrm{ver}}}^{\mathrm{ver}}(\pi_{V^{\mathrm{ver}}}(\boldsymbol{x})) + P_{G^{\mathrm{sel}}}^{\mathrm{sel}}(\pi_{V^{\mathrm{sel}}}(\boldsymbol{x})),$$

where $\pi_{V'}$ is the projection of the vector $\boldsymbol{x} \in \{0,1\}^n$ to the variables corresponding to the vertices in $V' \subseteq V$.

Glueability. We are going to solve graph problems, slicing the given graph into small pieces with the help of small separators. The separators will serve as boundaries between the different graph parts into which the graph is split. For each possible assignment of the vertices in the separators, we want to—independently—solve the corresponding problems on the graph parts and then reconstruct a solution for the whole graph by "gluing" together the solutions for the graph parts. We need to assign *colors* to the separator vertices in the course of the algorithm. Hence, our algorithm has to be designed in such a manner that it can also cope with colored graphs. In general (e.g., in the case of the dominating set problem), it is not sufficient to simply use the two colors 1 (for encoding "in the selected set") and 0 (for "not in the selected set").

Let us introduce some auxiliary notions. Let $G = (V, E)$ be an undirected graph and let C_0, C_1 be finite, disjoint sets. A C_0-C_1-*coloring* of G is a function $\chi : V \to C_0 + C_1 + \{\#\}$.[4] For $V' \subseteq V$, a function $\chi : V' \to C_0 + C_1$ can naturally be extended to a C_0-C_1-coloring of G by setting $\chi(v) = \#$ for all $v \in V \setminus V'$.

Consider a vector $\boldsymbol{x} \in \{0,1\}^{|V|}$. Let χ be a C_0-C_1-coloring of G. Then, $\boldsymbol{x}$ is *consistent* with χ, written $\boldsymbol{x} \sim \chi$, if, for $i = 0, 1$ and $j = 1, \ldots, |V|$, $\chi(v_j) \in C_i \Rightarrow x_j = i,.$

If χ is a C_0-C_1-coloring of G and if χ' is a C_0'-C_1'-coloring of G, then χ is *preserved* by χ', written $\chi \rightsquigarrow \chi'$, if $\forall v \in V \, \forall i \in \{0, 1\} \, (\chi(v) \in C_i \Rightarrow \chi'(v) \in C_i')$.

In the next section, when doing the divide and conquer approach with a given separator, we will deal with colorings on two different color sets: one color set $C^{\text{int}} := C_0^{\text{int}} + C_1^{\text{int}} + \{\#\}$ of *internal* colors that will be used for the assignments of colors to the separator vertices and a color set $C^{\text{ext}} := C_0^{\text{ext}} + C_1^{\text{ext}} + \{\#\}$ of *external* colors that will be used for handing down the information in the divide-step of the algorithm. The idea is that, in each recursive step, we will be confronted with a graph "pre-colored" with external colors. Every function $\oplus$ that assigns to a pair $(\chi^{\text{ext}}, \chi^{\text{int}})$ with $\chi^{\text{ext}} : V \to C^{\text{ext}}$, $\chi^{\text{int}} : V \to C^{\text{int}}$, $\chi^{\text{ext}} \rightsquigarrow \chi^{\text{int}}$, a $(C_0^{\text{ext}}$-$C_1^{\text{ext}})$-coloring $\chi^{\text{ext}} \oplus \chi^{\text{int}}$ is called a *recoloring* if $\chi^{\text{int}} \rightsquigarrow \chi^{\text{ext}} \oplus \chi^{\text{int}}$.

From the point of view of recursion, χ^{ext} is the pre-coloring which a certain recursion instance "receives" from the calling instance and χ^{int} represents a coloring which this instance assigns to a certain part of the graph. The coloring $\chi^{\text{ext}} \oplus \chi^{\text{int}}$ is handed down in the recursion.

We now introduce the central notion of "glueable" select&verify problems. This formalizes those problems that can be solved with separator based divide and conquer techniques as described above.

Definition 1. *A select&verify problem $\mathcal{G}$ given by $(P_\cdot, \mathrm{opt})$ is* glueable with σ colors *if there exist*

- *a color set $C^{int} := C_0^{int} + C_1^{int} + \{\#\}$ of internal colors with $|C_0^{int} + C_1^{int}| = \sigma$;*
- *a color set $C^{ext} := C_0^{ext} + C_1^{ext} + \{\#\}$ of external colors;*
- *a polynomial time computable function $h : (\mathbb{R}_+ \cup \{\pm\infty\})^3 \to \mathbb{R}_+ \cup \{\pm\infty\}$;*

and, for every n-vertex graph $G = (V, E)$ and subgraphs $G^{ver}, G^{sel} \subseteq G$ with a separation (A_1, S, A_2) of G^{ver}, we find

- *recolorings $\oplus_X$ for each $X \in \{A_1, S, A_2\}$, and*
- *for each internal coloring $\chi^{int} : S \to C^{int}$, subgraphs $G_{A_i}^{ver}(\chi^{int})$ of G^{ver} with $G^{ver}[A_i] \subseteq G_{A_i}^{ver}(\chi^{int}) \subseteq G^{ver}[A_i + S]$ for $i = 1, 2$, and subgraphs $G_S^{ver}(\chi^{int})$ of G^{ver} with $G_S^{ver}(\chi^{int}) \subseteq G^{ver}[S]$*

such that, for each external coloring $\chi^{ext} : V \to C^{ext}$,

$$\begin{aligned} &\mathrm{opt}\{P_{G^{ver}}(\boldsymbol{x} \mid G^{sel}) \mid \boldsymbol{x} \in \{0,1\}^n \wedge \boldsymbol{x} \sim \chi^{ext}\} \\ &= \mathrm{opt}_{\substack{\chi^{int}: S \to C_0^{int} + C_1^{int} \\ \chi^{ext} \rightsquigarrow \chi^{int}}} h\big(Eval_{A_1}(\chi^{int}), Eval_S(\chi^{int}), Eval_{A_2}(\chi^{int})\big). \end{aligned} \tag{1}$$

[4] The symbol # will be used for the undefined (i.e., not yet defined) color.

Here, $Eval_X(\cdot)$ for $X \in \{A_1, S, A_2\}$ is of the form $Eval_X(\chi^{int}) =$

$$\text{opt}\{P_{G_X^{ver}(\chi^{int})}(\boldsymbol{x} \mid G^{ver}[X] \cap G^{sel}) \mid \boldsymbol{x} \in \{0,1\}^n \wedge \boldsymbol{x} \sim (\chi^{ext} \oplus_X \chi^{int})\}.$$

Lemma 1. VERTEX COVER *and* INDEPENDENT SET *are glueable with 2 colors and* DOMINATING SET *is glueable with 4 colors.*

Proof. For VERTEX COVER, we use the color sets $C_i^\ell := \{i^\ell\}$ for $\ell \in \{\text{int}, \text{ext}\}$ and $i = 0, 1$. The function h is $h(x,y,z) = x+y+z$. The subgraphs $G_X^{\text{ver}}(\chi^{\text{int}})$ for $X \in \{A_1, S, A_2\}$ and $\chi^{\text{int}} : S \to C_0^{\text{int}} + C_1^{\text{int}}$ are $G_X^{\text{ver}}(\chi^{\text{int}}) := G^{\text{ver}}[X]$. In this way, the subroutine $\text{Eval}_S(\chi^{\text{int}})$ checks whether the coloring χ^{int} yields a vertex cover on $G^{\text{ver}}[S]$ and the subroutines $\text{Eval}_{A_i}(\chi^{\text{int}})$ compute the minimum size vertex cover on $G^{\text{ver}}[A_i]$. However, we still need to make sure that all edges going from A_i to S are covered. If a vertex in S is assigned a 1^{int} by χ^{int}, the incident edges are already covered. In the case of a 0^{int}-assignment for a vertex $v \in S$, we can color all neighbors in $N(v) \cap A_i$ to belong to the vertex cover. This is done by the following recolorings "$\oplus_{A_i}$." Define

$$(\chi^{\text{ext}} \oplus_{A_i} \chi^{\text{int}})(v) = \begin{cases} 0^{\text{ext}} \text{ if } \ \chi^{\text{int}}(v) = 0^{\text{int}}, \\ 1^{\text{ext}} \text{ if } \ \chi^{\text{int}}(v) = 1^{\text{int}} \text{ or } \exists w \in N(v) \text{ with } \chi^{\text{int}}(w) = 0^{\text{int}}, \\ \#, \ \ \text{otherwise.} \end{cases}$$

By this recoloring definition, an edge between a separator vertex and a vertex in A_i which is not covered by the separator vertex (due to the currently considered internal coloring) will be covered by the vertex in A_i. Our above reasoning shows that—with these settings—equation (1) is satisfied.

INDEPENDENT SET is shown to be glueable with 2 colors by a similar idea.

Regarding DOMINATING SET, we refer to the full paper. □

We want to mention in passing that—besides the problems given in Lemma 1—many more select&verify problems are glueable. In particular, this is true for the weighted versions and variations of the above mentioned problems.

4 Fixed parameter divide and conquer algorithms

For the considerations in this sections, let us fix a graph class $\mathbb{G}$ for which a $\sqrt{\cdot}$-separator theorem with constants α and β is known. Then, we consider a select&verify graph problem $\mathcal{G}$ defined by $(P_\cdot, \text{opt})$ that is glueable with σ colors.

4.1 Using glueability for divide and conquer

The evaluation of the term $\text{opt}_{\boldsymbol{x} \in \{0,1\}^n} P_G(\boldsymbol{x})$ can be done recursively as follows. Start the computation with $\text{opt}\{P_G(\boldsymbol{x}) \mid \boldsymbol{x} \in \{0,1\}^n\} = \text{opt}\{P_{G^{\text{ver}}}(\boldsymbol{x} \mid G^{\text{sel}}) \mid \boldsymbol{x} \in \{0,1\}^n,\ \boldsymbol{x} \sim \chi_0^{\text{ext}}\}$, where "$\chi_0^{\text{ext}} \equiv \#$" and $G^{\text{ver}} = G^{\text{sel}} = G$.

When $\text{opt}_{\boldsymbol{x} \in \{0,1\}^n,\, \boldsymbol{x} \sim \chi^{\text{ext}}} P_{G^{\text{ver}}}(\boldsymbol{x} \mid G^{\text{sel}})$ needs to be calculated for some $G^{\text{sel}}, G^{\text{ver}} \subseteq G$, and an external coloring $\chi^{\text{ext}} : V(G) \to C_0^{\text{ext}} + C_1^{\text{ext}} + \{\#\}$, we do the following:

1. If G^{ver} has size greater than some constant c, then find a $\sqrt{\cdot}$-separator for G^{ver} with $V(G^{\mathrm{ver}}) = A_1 + S + A_2$.
2. For all internal colorings χ^{int} of S with $\chi^{\mathrm{ext}} \rightsquigarrow \chi^{\mathrm{int}}$ do:
   (a) Determine $\mathrm{Eval}_{A_i}(\chi^{\mathrm{int}})$ recursively for $i = 1, 2$.
   (b) Determine $\mathrm{Eval}_S(\chi^{\mathrm{int}})$.
3. Return $\mathrm{opt}_{\chi^{\mathrm{int}},\chi^{\mathrm{ext}}\rightsquigarrow\chi^{\mathrm{int}}} h(\mathrm{Eval}_{A_1}(\chi^{\mathrm{int}}), \mathrm{Eval}_S(\chi^{\mathrm{int}}), \mathrm{Eval}_{A_2}(\chi^{\mathrm{int}}))$.

The size of the subproblems, i.e., the size of the graphs $G^{\mathrm{ver}}_{A_i}(\chi^{\mathrm{int}})$ which are used in the recursion, plays a crucial role in the analysis of the running time of this algorithm.

Definition 2. *A glueable select&verify problem is called* slim *if the subgraphs $G^{ver}_{A_i}(\chi^{int})$ are only by a constant number of vertices larger than $G^{ver}[A_i]$, i.e., if there exists a $\eta \geq 0$ such that $|V(G^{ver}_{A_i}(\chi^{int}))| \leq |A_i| + \eta$ for all internal colorings $\chi^{int} : S \to C^{\,int}$.*

Note that the proof of Lemma 1 shows that both VERTEX COVER and INDEPENDENT SET are slim with $\eta = 0$, whereas DOMINATING SET is not, as exhibited in the paper's long version. The following proposition gives the running time of the above algorithm. In order to assess the time required for the above given divide and conquer algorithm, we use the following abbreviations for the running times of certain subroutines: $T_S(n)$ denotes the time to find a separator in an n-vertex graph from class $\mathbb{G}$. $T_M(n)$ denotes the time to construct the modified graphs $G^{\mathrm{ver}}_X(\chi^{\mathrm{int}}) \in \mathbb{G}$ and the modified colorings $(\chi^{\mathrm{ext}} \oplus_X \chi^{\mathrm{int}})$ (for $X = \{A_1, S, A_2\}$ and each internal coloring χ^{int} with $\chi^{\mathrm{ext}} \rightsquigarrow \chi^{\mathrm{int}}$) from an n-vertex graph from class $\mathbb{G}$. $T_E(m)$ is the time to evaluate $\mathrm{Eval}_S(\chi^{\mathrm{int}})$ for all χ^{int}, $\chi^{\mathrm{ext}} \rightsquigarrow \chi^{\mathrm{int}}$, in a separator of size $m = \beta\sqrt{n}$. $T_g(n)$ is the time for gluing the results obtained by two sub-problems each of size $O(n)$. In the following, we assume that all these functions are polynomials.

Proposition 1. *For every $G \in \mathbb{G}$, $\mathrm{opt}_{\boldsymbol{x}\in\{0,1\}^n} P_G(\boldsymbol{x})$ can be computed in time*

$$c(\alpha', \beta, \sigma)^{\sqrt{n}} q(n), \quad \textit{where} \quad c(\alpha', \beta, \sigma) = \sigma^{\beta/(1-\sqrt{\alpha'})}.$$

Here, $\alpha' = \alpha + \epsilon$ for any $\epsilon \in (0, 1-\alpha)$ and the running time analysis only holds for $n \geq n_0(\epsilon)$, and q is some polynomial. If, however, $\mathcal{G}$ is slim or the $\sqrt{\cdot}$-separator theorem yields cycle separators, then the running time for the computation is $c(\alpha, \beta, \sigma)^{\sqrt{n}} q(n)$, which then holds for all n.

Proof. (Sketch) Let $T(n)$ denote the time to compute $\mathrm{opt}_{\boldsymbol{x}\in\{0,1\}^n,\, \boldsymbol{x}\sim\chi^{\mathrm{ext}}} P_{G^{\mathrm{ver}}}(\boldsymbol{x} \mid G^{\mathrm{sel}})$ for a graph $G^{\mathrm{ver}} = (V^{\mathrm{ver}}, E^{\mathrm{ver}})$ with $n = |V^{\mathrm{ver}}|$ (where $\chi^{\mathrm{ext}} : V(G) \to C^{\mathrm{ext}}_0 + C^{\mathrm{ext}}_1 + \{\#\}$ is some external coloring and $G^{\mathrm{sel}}, G^{\mathrm{ver}} \subseteq G$). In the case of the existence of a cycle separator theorem or if the problem is slim, the recurrence we have to solve in order to compute an upper bound on $T(n)$ then reads as follows:

$$T(n) \leq \sigma^{\beta\sqrt{n}} \cdot 2\,T(\alpha n) \cdot \underbrace{(T_M(n) + T_E(\beta\sqrt{n}) + T_g(\alpha n + \beta\sqrt{n}))}_{=:T_{M,E,g}(n)} + T_S(n).$$

The functions $T_{M,E,g}(n)$ and $T_S(n)$ are polynomials. The solution is given by $T(n) \leq \sigma^{(\beta/(1-\sqrt{\alpha}))\cdot\sqrt{n}} q(n)$ for some polynomial $q(\cdot)$. In the general case, from the definition of glueability, we have that the size of the two remaining subproblems to be solved recursively is $|V(G^{\text{ver}}_{A_i}(\chi^{\text{int}}))| \leq \alpha n + \beta\sqrt{n}$ for each χ^{int} that preserves χ^{ext}. Since with $\alpha' = \alpha + \epsilon$, for some $\epsilon \in (0, 1-\alpha)$, the inequality $\alpha n + \beta\sqrt{n} \leq \alpha' n$ holds for sufficiently large n, the result follows. □

4.2 How (linear) problem kernels help

Proposition 1 together with the existence of problem kernels yields:

Theorem 1. *Suppose that $\mathcal{G}$ admits a problem kernel of polynomial size $p(k)$ on $\mathbb{G}$ computable in time $T_K(n,k)$. Then, there is an algorithm to decide $(G,k) \in \mathcal{G}$, for a graph $G \in \mathbb{G}$, in time*

$$c(\alpha', \beta, \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k), \quad \textit{where} \quad c(\alpha', \beta, \sigma) = \sigma^{\beta/(1-\sqrt{\alpha'})}, \tag{2}$$

and $\alpha' = \alpha + \epsilon$ for any $\epsilon \in (0, 1-\alpha)$, holding only for $n \geq n_0(\epsilon)$, where $q(\cdot)$ is some polynomial. If, however, $\mathcal{G}$ is slim or the $\sqrt{\cdot}$-separator theorem yields cycle separators, then the running time for the computation is $c(\alpha, \beta, \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k)$, which then holds for all k. □

In particular, this means that for glueable select&verify problems for planar graphs that admit a *linear* problem kernel of size dk, we get an algorithm of running time $O\big(c(\alpha', \beta, \sigma, d)^{\sqrt{k}} q(k) + T_K(n,k)\big)$, where $c(\alpha', \beta, \sigma, d) = \sigma^{\beta\sqrt{d}/(1-\sqrt{\alpha'})}$.

Since VERTEX COVER is a slim problem, Theorem 1 yields a $c^{\sqrt{gk}}$-algorithm for $\mathbb{G}_g$, where $\mathbb{G}_g$ denotes the class of graphs of genus bounded by g, see [5].

4.3 Towards avoiding (linear) problem kernels: the core concept

We are going to introduce the novel notion of *problem cores*, which is closely related to that of problem kernels, but seemingly "incomparable" and tailored towards *unweighted minimization* select&verify problems. The idea is to restrict (only) the size of the "selection space", while—unlike in the setting of problem kernels—the whole (possibly large) problem instance may be still used for "checking".

Definition 3. *Consider an unweighted select&verify minimization graph problem $\mathcal{G}$ specified by $(P_\cdot, \min)$. A* corer of size $p(k)$ *is a polynomial time computable mapping $\phi : ((V,E),k) \mapsto V_c$ satisfying $|V_c| \leq p(k)$, and $\exists \boldsymbol{x} = (x_1, \ldots, x_{|V|}) \in \{0,1\}^{|V|}(P_G(\boldsymbol{x}) \leq k \wedge \{v_i \in V \mid x_i = 1\} \subseteq V_c)$. The set V_c is also called the* problem core *of $\mathcal{G}$. If $p(k) = ak$, we call ϕ a* linear *corer. In this case V_c is called a* factor-a problem core.

Note that weighted minimization problems could also be treated similarly at the expense of further technical complications. Having a problem core automatically makes a select&verify problem a "simple" one: For the problem core V_c, which can be computed in polynomial time, it is enough to check all k-element subsets, giving fixed parameter tractability. Stirling's formula yields:

Lemma 2. *If a select&verify problem $\mathcal{G}$ has a size ak problem kernel or if the core is a factor-a core, then there is a "$\min\{(ea)^k, 2^{ak}\}$-algorithm" for $\mathcal{G}$.* □

Even though there seems to be no *general* interrelation between problem kernels and cores, for our purposes, the different concepts can be interchanged:

Theorem 2. *Let G be an n-vertex-graph from a graph class $\mathbb{G}$ for which a $\sqrt{\cdot}$-separator theorem for* weighted graphs *is known with constants α, β. Suppose that $\mathcal{G}$ admits a corer of size $p(k)$, which can be computed in polynomial time $T_C(n)$. Then, there is an algorithm to decide $(G, k) \in \mathcal{G}$, for $G \in \mathbb{G}$, in time*

$$c(\alpha', \beta, \sigma)^{\sqrt{p(k)}} q(k) + T_C(n), \quad \text{where} \quad c(\alpha', \beta, \sigma) = \sigma^{\beta/(1-\sqrt{\alpha'})},$$

and $\alpha' = \alpha + \epsilon$ for any $\epsilon \in (0, 1 - \alpha)$, holding only for $n \geq n_0(\epsilon)$. If, however, $\mathcal{G}$ is slim or the $\sqrt{\cdot}$-separator theorem yields cycle separators, then the time for the computation is $c(\alpha, \beta, \sigma)^{\sqrt{p(k)}} q(k) + T_C(n)$, which holds for all k.

Proof. (Sketch) Consider $G = (V, E) \in \mathbb{G}$. The algorithm proceeds as follows:
(1) Compute a core $V_c \subseteq V$ containing at most $p(k)$ vertices in time $T_C(n)$.
(2) Then, $G[V_c] \in \mathbb{G}$ is the graph from which vertices have to be selected.
(3) Find an optimal $\boldsymbol{x}$ satisfying $P_G(\boldsymbol{x} \mid G[V_c])$ by applying the algorithm outlined before Prop. 1. One modification of this algorithm, however, is necessary: we do not use a separator theorem in step 1 of that algorithm for a separation of G^{ver}, but we add *weights* to G^{ver} as follows: V_c induces a weight function ω on G^{ver} by letting $\omega(v) = 1/|V_c|$, if $v \in V_c$ and $\omega(v) = 0$, otherwise. Then, we apply a separator theorem for weighted graphs to G^{ver} with the weight function ω. The constants $c(\alpha, \beta, \sigma)$ and $c(\alpha', \beta, \sigma)$ then are derived as in Prop. 1. □

5 Conclusion and further results

Summary. We exhibited how to use separator theorems for obtaining $c^{\sqrt{k}}$-fixed parameter divide and conquer algorithms. We defined "glueable select&verify problems," capturing graph problems such as VERTEX COVER and DOMINATING SET, as a problem class that allows for a divide and conquer approach on certain graph classes. Admittedly, the constants within these algorithms are still rather huge. For example, in the case of PLANAR VERTEX COVER, Theorem 1 yields a $37181^{\sqrt{k}}$-algorithm. In the long version, we elaborate on several ideas to overcome this weakness:

Further results (see the full version of the paper). Firstly, we analyze how Lipton and Tarjan proved their famous planar separator theorem; basically, the proof consists of two steps: in the first step, the given graph is thinned into pieces of "small" radius, and in the second step, a special (cycle) separator lemma for planar graphs with bounded radius is used. The such obtained separator therefore consists of two parts. Since after one application of Lipton and Tarjan's separator theorem (in a divide and conquer algorithm as described in Section 4), the remaining graph pieces still have relatively small radius, one could avoid the first thinning step. Iterating this idea in an optimized fashion, one gets, e.g., in the case of PLANAR VERTEX COVER, a $8564^{\sqrt{k}}$-algorithm. It is a challenge if better algorithm bounds are obtainable by using other separator theorems.

Secondly, we discuss the idea of stopping the recursion before having graph parts of constant size and then applying, e.g., elaborated search tree algorithms to these small parts. More precisely, a divide and conquer algorithm would once use Lipton and Tarjan's planar separator theorem for slicing the graph into pieces of small radius and then use only the mentioned special (cycle) separator lemma in the remainder of the recursion, until all graph pieces are sufficiently small. In this way, one gets $c^{k^{2/3}}$-algorithms with reasonable small constants c. For example, we derive a $7.7670^{k^{2/3}}$-algorithm for PLANAR VERTEX COVER.

Future research. We briefly sketch three lines of future research. (1) An alternative idea in order to lower the involved constants would be to devise new separator theorems with constants α and β, not only concentrating on bringing down β for fixed α (as, e.g., done for $\alpha = 2/3$ in [6]), but on minimizing the function $\beta/(1-\sqrt{\alpha})$. (2) It is an issue of future research to further investigate the newly introduced concept of cores. For example: is there a linear size core (or kernel) for PLANAR DOMINATING SET? (3) Finally, this paper (together with [1, 2]) might be a stimulus to investigate the structure "within" FPT in order to distinguish problems allowing for $c^{\sqrt{k}}$-algorithms from problems which seem to allow only for c^k-algorithms. Is it possible to develop a reasonable fine-grained structural theory of FPT? Note that Cai and Juedes [3] very recently showed that for a list of parameterized problems (e.g., for VERTEX COVER on general graphs) $c^{o(k)}$-algorithms are impossible unless FPT $= W[1]$.

References

1. J. Alber, H. L. 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.
2. J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. Technical Report TR01–023, ECCC Reports, Trier, March 2001. Extended abstract in *Proc. 28th ICALP*, Springer, 2001.
3. L. Cai and D. Juedes. Subexponential parameterized algorithms collapse the W-hierarchy. In *Proc. 28th ICALP*, Springer, 2001.
4. 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.
5. H. N. Djidjev. A separator theorem for graphs of fixed genus. *Serdica*, 11:319–329, 1985.
6. H. N. Djidjev and S. Venkatesan. Reduced constants for simple cycle graph separation. *Acta Informatica*, 34:231–243, 1997.
7. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.
8. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36:177–189, 1979.
9. R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9(3):615–627, 1980.