Journal of Computer and System Sciences, Vol. 67(4), pp. 808–832, 2003

Graph Separators: A Parameterized View*

Jochen Alber** Henning Fernau*** Rolf Niedermeier

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

Abstract. Graph separation is a well-known tool to make (hard) graph problems accessible to a divide-and-conquer approach. We show how to use graph separator theorems in combination with (linear) problem kernels 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 algorithm of running time $c^{\sqrt{k}} \cdot n^{O(1)}$ for constant c.
One of the main contributions of the paper is to exactly compute the base c of the exponential term and its dependence on the various parameters specified by the employed separator theorem and the underlying graph problem. We discuss several strategies to improve on the involved constant c.

Keywords: Planar graph problems, fixed parameter tractability, parameterized complexity, graph separator theorems, divide-and-conquer algorithms.

1 Introduction

It is a common fact that algorithm designers are often faced with problems that can be shown to be *NP*-hard [21]. 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 [6, 18, 20].

Fixed-parameter tractability. Formally, a parameterized problem is a (two-dimensional) language L over $\Sigma^* \times \mathbb{N}$, where Σ is some alphabet.[1] The second

* An extended abstract of this paper was presented at the 7th Annual International Computing and Combinatorics Conference (COCOON 2001), Springer-Verlag, LNCS 2108, pages 318–327, held in Guilin, China, August 2001.

** Supported by the Deutsche Forschungsgemeinschaft (research project PEAL (Parameterized complexity and Exact ALgorithms), NI 369/1-1,1-2).

*** Current affiliation: University of Newcastle, School of Electrical Engineering and Computer Science, University Drive, Callaghan, NSW 2308, Australia, fernau@cs.newcastle.edu.au

[1] In this paper, the parameter will always be a number. This number is encoded in unary, so that we need not distinguish between the parameter and its size.

coordinate of an element $(I,k) \in L$ is called the *parameter*. We say that L is *fixed-parameter tractable* if there exists an algorithm that decides the word problem on input (I,k) running in time $f(k)n^{O(1)}$, where $n = |I|$ and f is an arbitrary function that captures the inherent combinatorial explosion of the problem and that only depends on k. The associated complexity class is FPT. We will also term such algorithms "$f(k)$-algorithms" for brevity, focusing on the exponential part of the running time bound. Typically in the literature, such functions f for fixed-parameter problems are $f(k) = c^k$, $f(k) = k^k$, or $f(k) = c^{k^2}$. To our knowledge, so far, only one non-trivial fixed-parameter tractability result where the corresponding function f is sublinear in the exponent, namely $f(k) = c^{\sqrt{k}}$, is known [1] : Dominating Set on planar graphs. Similar results hold for closely related problems on planar graphs such as Face Cover, Independent Dominating Set, Weighted Dominating Set, etc. [1]. We improved this result to apply to a much broader class of planar graph problems, presenting a general methodology based on concepts such as tree decompositions and bounded outerplanarity and introducing the novel so-called "Layerwise Separation Property" as a key unifying tool [5]. The techniques there are rather elaborated. Here, by way of contrast, we will present a conceptually easier divide-and-conquer approach based on (planar) separator theorems for obtaining parameterized graph algorithms running in time $O(c^{e(k)} \cdot q(n))$ for sublinear functions e, i.e., $e(k) \in o(k)$. In contrast to [5], the techniques in this paper apply not only to planar graphs, but also to all classes of graphs which admit suitable separator theorems, such as, e.g., the class of graphs of bounded genus.

Scope of the paper. Up to now, several interesting but specialized fixed-parameter algorithms have been developed. The thrust has been to improve running times in a problem-specific manner, e.g., by extremely sophisticated case distinctions as can be seen in the case of Vertex Cover [12, 26, 27]. 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 [23], in combination with known algorithms for obtaining linear size problem kernels, in order to get fixed-parameter algorithms.

Our approach can be sketched as follows: We will apply the problem kernel reduction and, then, use (planar) separator theorems as already Lipton and Tarjan [24] did in order to pursue a divide-and-conquer strategy on the set of reduced instances. We did, however, take much more care for the impact of the "graph separator parameters" on the recurrences in the running time analysis. In addition, we obviously consider a broader class of problems that can be attacked by this approach (namely, in principle, all so-called glueable select&verify problems such as, e.g., Dominating Set[2]). Doing so, we exhibit the importance of a special form of separators, so-called cycle separators and their influence on

[2] Lipton and Tarjan only describe in details a solution for the structurally much simpler Independent Set.

the running time analysis. Moreover, we show how to employ different separator-finding strategies in order to get divide-and-conquer algorithms with constants which are better than those corresponding to a direct use of the best known (planar) separator theorems. Finally, we discuss possible combinations of separator techniques with other solution methods (like search tree based algorithms), typically leading to $c^{k^{2/3}}$-algorithms for problems with linear kernels. The running time of our algorithms is mostly bounded by $c^{\sqrt{k}}q(n)$ or by $c^{k^{2/3}}q(n)$ for some constant $c > 1$ and some polynomial $q(\cdot)$.[3] Although the constants achieved in our setting so far seem to be mostly too large in order to yield practical worst-case algorithms, it provides a general and sound mathematical formalization of a rich class of problems that allow for divide-and-conquer solutions based on separators and it provides a strong link to fixed-parameter tractability.

2 Basic definitions and preliminaries

We consider undirected graphs $G = (V, E)$, where V denotes the vertex set and E denotes the edge set. In our setting, all graphs are simple (i.e., with no double edges) without loops. Sometimes, we refer to V by $V(G)$; by $N(v)$ we refer to the set of vertices adjacent to v; $G[D]$ denotes the subgraph induced by 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 with vertex set $V_1 \cap V_2$ and edge set $E_1 \cap E_2$. A graph $G' = (V', E')$ is a subgraph of $G = (V, E)$, denoted by $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. The *radius* of a graph G is the minimal height of a rooted spanning tree of G.

In this paper, we only consider graph classes, denoted by $\mathbb{G}$, that are closed under taking subgraphs. The most important among the graph classes considered in this paper is that of *planar* graphs, i.e., graphs that have a drawing in the plane without edge crossings. A *plane* graph, or a *planar embedding,* is any drawing of a planar graph in the plane without edge crossing.

A *parameterized graph problem* is a language consisting of tuples (G, k), where G is a graph and k is an integer. A *parameterized graph problem on planar graphs* is a parameterized graph problem, where the graph G of an instance (G, k) is assumed to be planar.

If $G = (V, E)$ is a planar graph, a *triangulation* $\hat{G} = (V, \hat{E})$, where $E \subseteq \hat{E}$, of G is a planar graph such that, for any additional edge $\{v, v'\} \notin \hat{E}$ with $v, v' \in V$, $(V, \hat{E} \cup \{v, v'\})$ is not planar. The *faces* of a plane graph are the maximal regions of the plane that contain no point used in the embedding. Among others, we study the following "graph numbers" $vc(\cdot)$, $is(\cdot)$, and $ds(\cdot)$:

- A *vertex cover* C of 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 with a minimum number of vertices is denoted by $vc(G)$.
- An *independent set* of G is a set of pairwise nonadjacent vertices; the size of an independent set with a maximum number of vertices is denoted by $is(G)$.

[3] Actually, whenever we can construct a so-called problem kernel of size $k^{O(1)}$ in polynomial time, then we can replace the term $c^{\sqrt{k}}n^{O(1)}$ by $c^{\sqrt{k}}k^{O(1)} + n^{O(1)}$.

- A *dominating set* D of G is a set of vertices such that each of the rest of the vertices in G 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 denoted by Vertex Cover, Independent Set, and Dominating Set.

Finally, to simplify notation, we write $A_1 + \cdots + A_n$ to denote the pairwisely disjoint union of sets A_1 through A_n. By $\mathbb{R}$ (and related notations) within algorithmic considerations, we refer to a "reasonably computable subset" of the real numbers. Sometimes, we extend real addition and multiplication to $\pm\infty$ in order to cover failure situations. Then, we follow the *convention* that $0 \cdot (\pm\infty) = 0$.

2.1 Linear problem kernels

Compared with other definitions of problem kernels, we are a bit more restrictive and precise in the following definition.

Definition 2.1. *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).* Reduction to problem kernel*, then, means to replace instance (I, k) by a "reduced" instance (I', k') (which we call* problem kernel*) such that*

$$k' \leq c \cdot k, \qquad |I'| \leq p(k)$$

with a constant c,[4] *some function p only depending on k, and*

$$(I, k) \in \mathcal{L} \text{ iff } (I', k') \in \mathcal{L}.$$

Furthermore, we require that the reduction from (I, k) to (I', k') is computable in polynomial time $T_K(|I|, k)$. The size *of the problem kernel is given by $p(k)$.*

Often (cf. the subsequent example Vertex Cover), the best one can hope for is that the problem kernel has size linear in k, a so-called *linear problem kernel*. For instance, using a theorem of Nemhauser and Trotter [25], (also cf. [10, 28]), Chen *et al.* [12] recently observed a problem kernel of size $2k$ for Vertex Cover on general (not necessarily planar) graphs.

By making use of the four color theorem for planar graphs and its corresponding algorithm generating a four coloring [30], it easily follows [5] that Independent Set on planar graphs has a problem kernel of size $4k$.

Recently, a linear problem kernel was proven for Dominating Set on planar graphs [3]. The worst-case upper bound on the kernel size there is $335k$.

Besides the positive effect of reducing the input size significantly and all obvious consequences of that, this paper gives further justification, in particular, for the importance of size $O(k)$ problem kernels. The point is that, once

[4] Usually, $c \leq 1$. In general, it would even be allowed that $k' = g(k)$ for some arbitrary function g. For our purposes, however, we need that k and k' are linearly related. We are not aware of a concrete, natural parameterized problem with problem kernel where this is not the case.

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 [23, 24]. The constant factor in the problem kernel size directly influences the value of the exponential base. Hence, lowering the kernel size means improved efficiency.

2.2 Classical separator theorems

Definition 2.2. *Let $G = (V, E)$ be an undirected graph. A* separator $S \subseteq V$ *of G divides V into two* parts $A_1 \subseteq V$ *and $A_2 \subseteq V$ such that*[5]

- *$A_1 + S + A_2 = V$, and*
- *no edge joins vertices in A_1 and A_2.*

Later, we will write δA_1 (or δA_2) as shorthand for $A_1 + S$ (or $A_2 + S$, respectively). The triple (A_1, S, A_2) is also called a separation *of G.*

Clearly, this definition can be generalized to the case where a separator partitions the vertex set into ℓ subsets instead of only two. We refer to such separators simply by *ℓ-separator.* The techniques we develop here all are based on the existence of "small" graph separators, which means that $|S|$ is bounded by $o(|V|)$.

Definition 2.3. *According to Lipton and Tarjan [23], an* $f(\cdot)$-separator theorem (with constants $\alpha < 1$, $\beta > 0$) *for a class $\mathbb{G}$ of graphs which is closed under taking vertex-induced 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.*

Again, this definition easily generalizes to ℓ-separators with $\ell > 2$.

Stated in this framework, the planar separator theorem due to Lipton and Tarjan [23] is a $\sqrt{\cdot}$-separator theorem with constants $\alpha = 2/3$ and $\beta = 2\sqrt{2} \approx 2.83$. The current record for $\alpha = 2/3$ is $\beta = \sqrt{2/3} + \sqrt{4/3} \approx 1.97$ [17]. Djidjev has also shown a lower bound of $\beta \approx 1.55$ for $\alpha = 2/3$ [15]. For $\alpha = 1/2$, the "record" is $\beta = 2\sqrt{6} \approx 4.90$ [7, 11]. A lower bound of $\beta \approx 1.65$ is known in this case [32]. For $\alpha = 3/4$, the best known value for β is $\sqrt{2\pi/\sqrt{3}} \cdot (1 + \sqrt{3})/\sqrt{8} \approx 1.84$ with a known lower bound of $\beta \approx 1.42$, see [32]. The results are summarized in Table 1.

In order to develop a flexible framework, we will do our calculations below always with the parameters α and β left unspecified up to the point where we try to give concrete numbers in the case of VERTEX COVER on planar graphs, which will serve as a running example. Also, we point out how the existence of ℓ-separators for $\ell > 2$ might improve the running time. In principle, our results also apply to graph problems for graphs from other graph classes with $\sqrt{\cdot}$-separator theorems as listed above. As indicated in [29], separator based techniques can be also used to solve counting problems instead of decision problems.

[5] In general, of course, A_1, A_2 and S will be non-empty. In order to cover boundary cases in some considerations below, we did not put this into the separator definition.

β	$\alpha = \frac{2}{3}$		$r(\frac{2}{3}, \beta)$	$\alpha = \frac{1}{2}$		$r(\frac{1}{2}, \beta)$	$\alpha = \frac{3}{4}$		$r(\frac{3}{4}, \beta)$
upper bounds	$\sqrt{\frac{2}{3}} + \sqrt{\frac{4}{3}}$	[17]	10.74	$\sqrt{24}$	[7, 11]	16.73	$\sqrt{\frac{2\pi}{\sqrt{3}} \cdot \frac{1+\sqrt{3}}{\sqrt{8}}}$	[32]	13.73
lower bounds	1.55	[15]	8.45	1.65	[32]	5.63	1.42	[32]	10.60

Table 1. Summary of various $\sqrt{\cdot}$-separator theorems with their constants α and β. Here, $r(\alpha, \beta)$ denotes the ratio $r(\alpha, \beta) = \beta/(1 - \sqrt{\alpha})$, which is of central importance to the running time analysis of our algorithms, cf. Proposition 4.3.

Variants of separator theorems

Cycle separators. In the literature, there are many separator theorems for planar graphs which guarantee that all the vertices of the separator lie on a simple cycle, provided that the given graph is biconnected or even triangulated. In fact, the current "record holder" in the case of $\alpha = 2/3$ yields a cycle separator, see [17]. From an algorithmic perspective, as explained below, the requirements of having biconnected or triangulated graphs are rarely met: even if the original graph was biconnected or triangulated, subgraphs which are obtained by recursive applications of separator theorems to a larger graph are not biconnected or triangulated in general. Therefore, we consider the following definition appropriate for our purposes:

Definition 2.4. *We will call a separator S of a planar graph G* cycle separator *if there exists a triangulation $\hat{G}$ of G such that S forms a simple cycle in $\hat{G}$.*

Note: some triangulation of a given planar graph can be computed in linear time.

Remark 2.5. It will turn out that it is of special value (concerning the design of divide and conquer algorithms) to have separators that form simple cycles (within some triangulation of the given graph G), since then the Jordan curve theorem applies (for planar graphs), which basically means that the separator S splits G into an "inside"-part A_1 and an "outside"-part A_2. If G is a subgraph of a larger planar graph $\tilde{G}$, then this implies that each vertex v of $\tilde{G}$ that has neighbors in A_1 has no neighbors in A_2 and vice versa. This observation is important, since it means that a local property pertaining to vertex v of $\tilde{G}$ (like: v belongs to a dominating set or not) can only influence vertices in δA_1 or vertices in δA_2.

Weighted separation. It is also possible to incorporate *weights* in most separator theorems. For our purposes, weights are nonnegative reals assigned to the vertices in a graph such that the sum of all weights in a graph is bounded by one. For weighted graphs, an $f(\cdot)$-separator theorem with constants α and β for graph class $\mathbb{G}$ guarantees, for any n-vertex graph $G \in \mathbb{G}$, the existence of a separation (A_1, S, A_2) of G such that

- neither A_1 nor A_2 has weight more than α, and
- S contains no more than $\beta f(n)$ vertices.

Other graph classes with separator theorems. Similar to the case of planar graphs, $\sqrt{\cdot}$-separator theorems are also known for other graph classes, e.g., for the class of graphs of bounded genus, see [16]. More generally, Alon, Seymour and Thomas proved a $\sqrt{\cdot}$-separator theorem for graph classes with an excluded complete graph minor [8, 9]. There are also interesting graph classes which are not minor closed for which $\sqrt{\cdot}$-separator theorems are known, as, e.g., the classes of ℓ-map graphs, see [13, 14].[6] Many comments of this paper apply to these more general situations, too.

3 Glueable graph problems

Based on the notion of separators, we will give a characterization of a whole class of problems that can be attacked by the approach that will be described in the subsequent sections. To this end, we coin the notions of select&verify graph problems and glueability. These notions are central to this paper. In the companion paper [5], we also introduce select&verify graph problems. Since the algorithms from [5] are not recursive, only a simplified notion of glueability (termed "weak glueability") is needed there.

3.1 Select&verify graph problems

Definition 3.1. *A set $\mathcal{G}$ of tuples (G, k), G an undirected graph with vertex set $V = \{v_1, \ldots, v_n\}$ and k a positive 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 $P_G = P_G^{sel} + P_G^{ver}$, where*

$$P_G^{sel} : \{0,1\}^n \to \mathbb{R}_+, \text{ (also called selecting function)}$$
$$P_G^{ver} : \{0,1\}^n \to \{0, \pm\infty\}, \text{ (also called verifying function)}^7 \text{ and}$$

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

For $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$, 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 $P_G^{ver}(\boldsymbol{x}) = 0$.*

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. The "numbers" $\pm\infty$ indicate the non-admissibility of a candidate solution.

[6] The ℓ basically refers to the maximal size of a complete subgraph of a map graph.

[7] More precisely, the range of a verifying function is $\{0, \infty\}$ in the case of a minimization problem and $\{0, -\infty\}$ in the case of a maximization problem.

Remark 3.2. Every select&verify graph problem that additionally admits a problem kernel of size $p(k)$ is solvable in time $O(2^{p(k)}p(k) + T_K(n, k))$.

Example 3.3. We now give some examples for select&verify problems by specifying the function $P_G = P_G^{\text{sel}} + P_G^{\text{ver}}$. In all cases below, the selecting function P_G for a graph $G = (V, E)$ will be

$$P_G^{\text{sel}}(\boldsymbol{x}) = \sum_{v_i \in V} x_i.$$

1. In the case of Vertex Cover, we have opt = min and use

   $$P_G^{\text{ver}}(\boldsymbol{x}) = \sum_{\{v_i,v_j\} \in E} \infty \cdot (1 - x_i)(1 - x_j),$$

   where this sum brings $P_G(\boldsymbol{x})$ to infinity whenever there is an uncovered edge. In addition, $P_G(\boldsymbol{x}) \leq k$ then guarantees a vertex cover set of size at most k. Clearly, P_G is polynomial time computable.
2. Similarly, in the case of Independent Set, we let opt = max and choose

   $$P_G^{\text{ver}}(\boldsymbol{x}) = \sum_{\{v_i,v_j\} \in E} -\infty \cdot x_i \cdot x_j.$$

3. In the case of Dominating Set, we have

   $$P_G^{\text{ver}}(\boldsymbol{x}) = \sum_{v_i \in V} (\infty \cdot (1 - x_i) \cdot \prod_{\{v_i,v_j\} \in E} (1 - x_j)),$$

   where this sum brings $P_G(\boldsymbol{x})$ to infinity whenever there is a non-dominated vertex which is not in the selected dominating set. In addition, $P_G(\boldsymbol{x}) \leq k$ then guarantees a dominating set of size at most k.
4. Similar observations as for Vertex Cover, Independent Set, and Dominating Set do hold for many other graph problems and, in particular, weighted variants of these.[8] As a source of problems, consider the variants of Dominating Set listed in [33–35]. In particular, the Total Dominating Set problem is defined by

   $$P_G^{\text{ver}}(\boldsymbol{x}) = \sum_{v_i \in V} \big(\infty \cdot \prod_{\{v_i,v_j\} \in E} (1 - x_j)\big).$$

Moreover, graph problems where a small (or large) *edge set* is sought for can often be reformulated into vertex set optimization problems by introducing an additional artificial vertex on each edge of the original graph. In this way, the NP-complete Edge Dominating Set [36] problem can be handled. Similarly, planar graph problems where a small (or large) *face set* is looked for are expressible as select&verify problems of the dual graphs or by introducing additional "face vertices."

[8] In the weighted case, one typically chooses a selecting function of the form $P_G^{\text{sel}}(\boldsymbol{x}) = \sum_{v_i \in V} \alpha_i x_i$, where α_i is the weight of the vertex v_i.

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.

Definition 3.4. *Let $P_{\cdot} = P_{\cdot}^{sel} + P_{\cdot}^{ver}$ be the function of a select&verify problem. For an n-vertex graph G and subgraphs $G^{ver} = (V^{ver}, E^{ver})$, $G^{sel} = (V^{sel}, E^{sel}) \subseteq G$, we let*

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

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

3.2 Glueability

We are going to solve graph problems recursively, slicing the given graph into small pieces with the help of small separators. Within these separators, the basic strategy will be to test all possible assignments of the vertices. For example, in the case of Vertex Cover, this means that, for a separator S, all possible functions $S \to \{0,1\}$ are tested, where assigning the "color" 0 means that the corresponding vertex is *not* in the (partial) cover and assigning 1 means that the corresponding vertex lies in the (partial) cover. In the case of more involved problems like (variants of) Dominating Set, more sophisticated assignments are necessary, as detailed below. The separators bound 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 remaining graph parts and then reconstruct a solution for the whole graph by "gluing" together the solutions for the graph parts. In order to do so, all additional information necessary for solving the subproblems correctly has to be transported and coded within the separators. It turns out that the information to be handed on is pretty clear in the case of Vertex Cover, but it is much more involved in the case of Dominating Set and many others. This is the basic motivation for the formal framework we develop in this subsection. 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, even though the original problem may have been a problem on non-colored graphs. In general (e.g., in the case of Dominating Set), 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"). This is why the set of colors will be some union of finite sets $C_0 + C_1$, instead of $\{0,1\}$ only. The usefulness of considering a colored version of "normal" graph problems is also testified in [2], where a colored version of Dominating Set on planar graphs was employed for developing a parameterized search tree algorithm.

Definition 3.5. *Let $G = (V,E)$ be an undirected graph and C_0, C_1 be finite, disjoint sets. A C_0-C_1-*coloring *of G is a function $\chi : V \to C_0 + C_1 + \{\#\}$.*

The symbol # will be used for the undefined (i.e., not yet defined) color. This means that, 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 $v \in V \setminus V'$.

Definition 3.6. *Consider an instance (G, k) of a select&verify problem $\mathcal{G}$ and a vector $\boldsymbol{x} \in \{0,1\}^n$ with $V(G) = \{v_1, \ldots, v_n\}$. Let χ be a C_0-C_1-coloring of G. Then, $\boldsymbol{x}$ is* consistent *with χ, written $\boldsymbol{x} \sim \chi$, if*

$$\chi(v_j) \in C_i \ \Rightarrow \ x_j = i, \quad \textit{for}\, i = 0, 1,\ j = 1, \ldots, n.$$

A divide-and-conquer algorithm will deal with colorings on two color sets: one color set $C^{\text{int}} := C_0^{\text{int}} + C_1^{\text{int}} + \{\#\}$ of *internal* colors used for the separator assignments and a color set $C^{\text{ext}} := C_0^{\text{ext}} + C_1^{\text{ext}} + \{\#\}$ of *external* colors 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 "precolored" with external colors. Our algorithm then finds a new separator and assigns internal colors to the vertices of this separator. These assignments of internal colors should be, in some sense, "compatible" with the precolored graph. Moreover, we want to be able to "recolor" the graph for the next divide-step. That means that we somehow have to be able to merge an external coloring (from the precolored graph) with an internal coloring (i.e., an assignment of our current separator) in a way such that we obtain a new (compatible) external coloring that can be handed down in the next recursive step.

Definition 3.7. *Let $G = (V, E)$ be a graph and let C_0^{int}, C_1^{int} and C_0^{ext}, C_1^{ext} be mutually disjoint, finite sets. Let $C^{int} := C_0^{int} + C_1^{int} + \{\#\}$ and let $C^{ext} := C_0^{ext} + C_1^{ext} + \{\#\}$.*

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*

$$\forall v \in V\, \forall i = 0, 1\, (\chi(v) \in C_i \Rightarrow \chi'(v) \in C_i').$$

Every function $\oplus$ that assigns a $(C_0^{ext}$-$C_1^{ext})$-coloring $\chi^{ext} \oplus \chi^{int}$ to a pair (χ^{ext}, χ^{int}) with $\chi^{ext} : V \to C^{ext}$, $\chi^{int} : V \to C^{int}$, and $\chi^{ext} \rightsquigarrow \chi^{int}$, is called a recoloring *if $\chi^{int} \rightsquigarrow \chi^{ext} \oplus \chi^{int}$.*

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 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. The notions $\chi^{\text{ext}} \rightsquigarrow \chi^{\text{int}}$ and $\chi^{\text{int}} \rightsquigarrow \chi^{\text{ext}} \oplus \chi^{\text{int}}$ express that any vector $\boldsymbol{x} \in \{0,1\}^{|V|}$ that is consistent with χ^{ext} is also consistent with the colorings χ^{int} and $\chi^{\text{ext}} \oplus \chi^{\text{int}}$.

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. We apply this rather abstract notion to concrete graph problems afterwards (Lemma 3.9). The following definition is best understood with the help of a concrete example as given in the proof of Lemma 3.9.

Definition 3.8. *A select&verify problem $\mathcal{G}$ given by $(P_\cdot, \text{opt})$ 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 if, 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* $\chi^{int} : S \to C_0^{int} + C_1^{int}$*,*
  subgraphs $G^{ver}_{A_i}(\chi^{int})$ *of* G^{ver} *with* $G^{ver}[A_i] \subseteq G^{ver}_{A_i}(\chi^{int}) \subseteq G^{ver}[\delta A_i]$ *for* $i = 1, 2$*, and*
  subgraphs $G^{ver}_S(\chi^{int})$ *of* G^{ver} *with* $G^{ver}_S(\chi^{int}) \subseteq G^{ver}[S]$

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

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

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

$$Eval_X(\chi^{int}) \quad = \quad \operatorname{opt}_{\substack{\boldsymbol{x}\in\{0,1\}^n \\ \boldsymbol{x}\sim(\chi^{ext}\oplus_X\chi^{int})}} P_{G^{ver}_X(\chi^{int})}(\boldsymbol{x} \mid G^{ver}[X] \cap G^{sel}). \tag{2}$$

Lemma 3.9. VERTEX COVER *and* INDEPENDENT SET *are glueable with 2 colors and* DOMINATING SET *is glueable with 4 colors.*

Proof. For VERTEX COVER (see Example 3.3.1), 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 $G^{\text{ver}}_X(\chi^{\text{int}})$ for $X \in \{A_1, S, A_2\}$ and $\chi^{\text{int}} : S \to C_0^{\text{int}} + C_1^{\text{int}}$ are $G^{\text{ver}}_X(\chi^{\text{int}}) := G^{\text{ver}}[X]$. 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

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

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 covering) will be covered by the vertex in A_i. Our above reasoning shows that—with these settings—Equation (1) in Definition 3.8 is satisfied.

INDEPENDENT SET (see Example 3.3.2)) is shown to be glueable with 2 colors by a similar idea.

To show that Dominating Set (see Example 3.3.3)) is glueable with 4 colors, we use the following color sets

$$C_0^{\text{int}} := \{0_{A_1}^{\text{int}}, 0_{A_2}^{\text{int}}, 0_S^{\text{int}}\},\ C_1^{\text{int}} := \{1^{\text{int}}\},$$
$$C_0^{\text{ext}} := \{0^{\text{ext}}\},\qquad C_1^{\text{ext}} := \{1^{\text{ext}}\}.$$

The semantics of these colors is as follows. Assigning the color 0_X^{int}, for $X \in \{A_1, A_2, S\}$, to vertices in a current separation $V = A_1 + S + A_2$ means that the vertex is not in the dominating set and will be dominated by a vertex in X. Clearly, 1^{int} will mean that the vertex belongs to the dominating set. The external colors simply hand down the information whether a vertex belongs to the dominating set, represented by 1^{ext}, or whether it is not in the dominating set *and* still needs to be dominated, represented by 0^{ext}.[9] The function h simply is addition, i.e., $h(x, y, z) = x + y + z$. When handing down the information to the subproblems, for a given internal coloring $\chi^{\text{int}} : S \to C_0^{\text{int}} + C_1^{\text{int}}$, we define

$$G_{A_i}^{\text{ver}}(\chi^{\text{int}}) := G^{\text{ver}}[A_i \cup (\chi^{\text{int}})^{-1}(\{1^{\text{int}}, 0_{A_i}^{\text{int}}\})] \quad \text{and}$$
$$G_S^{\text{ver}}(\chi^{\text{int}}) := G^{\text{ver}}[(\chi^{\text{int}})^{-1}(\{1^{\text{int}}, 0_S^{\text{int}}\})].$$

The recolorings $\oplus_X$ for $X \in \{A_1, S, A_2\}$ are chosen to be

$$(\chi^{\text{ext}} \oplus_X \chi^{\text{int}})(v) = \begin{cases} 0^{\text{ext}} \text{ if } \chi^{\text{int}}(v) \in C_0^{\text{int}}, \\ 1^{\text{ext}} \text{ if } \chi^{\text{int}}(v) = 1^{\text{int}}, \\ \#, \text{ otherwise.} \end{cases}$$

Let us explain in a few lines why—with these settings—Equation (1) in Definition 3.8 is satisfied. If an internal coloring χ^{int} assigns color 0_X^{int} ($X \in \{A_1, S, A_2\}$) to a vertex in S, then this vertex needs to be dominated by a neighbor in X. This will be checked in $\text{Eval}_X(\chi^{\text{int}})$ using the graph $G_X^{\text{ver}}(\chi^{\text{int}})$. To this end, vertices assigned the color 0_X^{int} (i.e., the set $(\chi^{\text{int}})^{-1}(\{0_X^{\text{int}}\})$) are included in $G_X^{\text{ver}}(\chi^{\text{int}})$. The vertices assigned color 1^{int} (i.e., $(\chi^{\text{int}})^{-1}(\{1^{\text{int}}\})$) also need to be handed down to the subroutines, since such a vertex may already dominate vertices in X. The recolorings merge the given external coloring χ^{ext} with the current internal coloring χ^{int} in a way that already assigned colors from C_i^{int} or C_i^{ext} ($i = 0, 1$) become i^{ext}. The terms $\text{Eval}_{A_i}(\chi^{\text{int}})$ then compute (for each internal coloring χ^{int}) the size of a minimum dominating set in A_i under the constraint that some vertices in δA_i still need to be dominated (namely, the vertices in $\delta A_i \cap (\chi^{\text{ext}} \oplus_{A_i} \chi^{\text{int}})^{-1}(0^{\text{ext}})$) and some vertices in δA_i can already be assumed to be in the dominating set (namely, the vertices in $\delta A_i \cap (\chi^{\text{ext}} \oplus_{A_i} \chi^{\text{int}})^{-1}(1^{\text{ext}})$). The term $\text{Eval}_S(\chi^{\text{int}})$ checks the correctness of the internal coloring χ^{int} of S. □

Note that, from the point of view of divide-and-conquer algorithms, three colors are enough for Dominating Set, since the color 1^{int} already determines the color 0_S^{int}. This issue is presented in more details in [5].
We illustrate the ideas of the dominating set algorithm by using an example.

[9] A vertex that is not in the dominating set but is already guaranteed to be dominated, e.g., by a vertex in the current separator, will never be handed down, since these vertices are of no use in the sequel of the recursion.

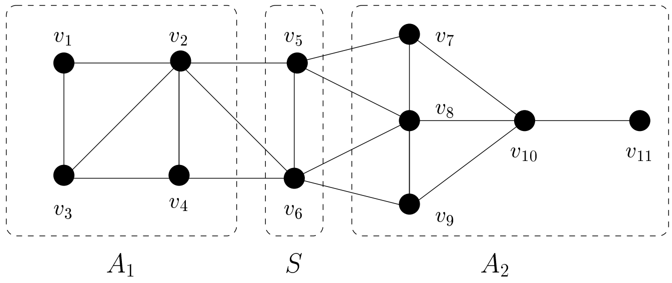

Fig. 1. A partitioned graph.

Example 3.10. Consider DOMINATING SET for the separated graph in Fig. 1. Beginning with the external coloring $\chi^{\text{ext}} \equiv \#$ and $G^{\text{ver}} = G^{\text{sel}} = G$, we need to go over all $4^2 = 16$ internal colorings $\chi^{\text{int}} : S \to \{0^{\text{int}}_{A_1}, 0^{\text{int}}_S, 0^{\text{int}}_{A_2}, 1^{\text{int}}\}$ (which trivially satisfy $\chi^{\text{ext}} \rightsquigarrow \chi^{\text{int}}$). As an example, we choose χ^{int} with $v_5 \mapsto 0^{\text{int}}_{A_1}$, $v_6 \mapsto 0^{\text{int}}_{A_2}$. In this case, we get $G^{\text{ver}}_{A_1}[\chi^{\text{int}}] = G^{\text{ver}}[\{v_1, \ldots, v_5\}]$, $G^{\text{ver}}_S[\chi^{\text{int}}] = \emptyset$, and $G^{\text{ver}}_{A_2}[\chi^{\text{int}}] = G^{\text{ver}}[\{v_6, \ldots, v_{11}\}]$. Recursively, we will use these graphs for the verifying function and the graphs $G^{\text{ver}}[A_1]$, $G^{\text{ver}}[S]$, and $G^{\text{ver}}[A_2]$ for the selecting function. The external colorings that will be handed down to the subproblems after the recoloring look as follows: On the graph $G^{\text{ver}}_{A_1}[\chi^{\text{int}}]$, we have $\chi_1(v_i) := (\chi^{\text{ext}} \oplus_{A_1} \chi^{\text{int}})(v_i) = \#$ for $i = 1, \ldots, 4$, and $\chi_1(v_5) = 0^{\text{ext}}$. On the graph $G^{\text{ver}}_{A_2}[\chi^{\text{int}}]$, we have $\chi_2(v_i) := (\chi^{\text{ext}} \oplus_{A_2} \chi^{\text{int}})(v_i) = \#$ for $i = 7, \ldots, 11$, and $\chi_2(v_6) = 0^{\text{ext}}$. Clearly,

$$\text{Eval}_{A_1}(\chi^{\text{int}}) = 2, \quad \text{Eval}_S(\chi^{\text{int}}) = 0, \quad \text{and} \quad \text{Eval}_{A_2}(\chi^{\text{int}}) = 2.$$

The minimum in Equation (2) of Definition 3.8 for $X = A_1$ is obtained, e.g., by choosing the vertices v_1 and v_2 (note that the latter needs to be chosen, since $\chi_1(v_5) = 0^{\text{ext}}$, meaning that v_5 is forced to be dominated in this term). The minimum for A_2 is obtained, e.g., by choosing the vertices v_8 and v_{10} (again, $\chi_2(v_6) = 0^{\text{ext}}$ forces either v_8 or v_9 to be in the dominating set). Hence,

$$h(\text{Eval}_{A_1}(\chi^{\text{int}}), \text{Eval}_S(\chi^{\text{int}}), \text{Eval}_{A_2}(\chi^{\text{int}})) = 2 + 0 + 2 = 4.$$

We obtain an optimal result, e.g., for the choice of the internal coloring χ^{int} with $\chi^{\text{int}}(v_5) = \chi^{\text{int}}(v_6) = 0^{\text{int}}_{A_2}$. Here, we get $\text{Eval}_{A_1}(\chi^{\text{int}}) = 1$, $\text{Eval}_S(\chi^{\text{int}}) = 0$, and $\text{Eval}_{A_2}(\chi^{\text{int}}) = 2$, for the possible choices of v_3, v_8, v_{10} as dominating set vertices.

Remark 3.11. In the case that recurrences are based on separator theorems yielding ℓ-separators, let us call a problem *ℓ-glueable with σ_ℓ colors* if σ_ℓ colors are to be distinguished in the recursion. For example, an extension of our previous lemma shows that DOMINATING SET is ℓ-glueable with $\ell+2$ colors, where $|C^{\text{int}}_0| = \ell + 1$. There need not be, however, a dependence of the number of colors on the number of graph parts: both Vertex Cover and Independent Set are ℓ-glueable with two colors.

Besides the problems stated in the preceding Lemma 3.9, many more select&verify problems are glueable, for example, those which are listed in [33–35]. In particular, weighted versions and variations of the problems discussed in Lemma 3.9 are glueable. Total Dominating Set is an example of a graph problem where a color set C_1^{int} of more than one color is needed.

4 Fixed-parameter divide-and-conquer algorithms

In this section, we provide the basic framework for deriving fixed-parameter algorithms based on the concepts we introduced so far.

4.1 Using glueability for divide-and-conquer

Fix a graph class $\mathbb{G}$ for which a $\sqrt{\cdot}$-separator theorem with constants α and β (cf. Definition 2.3) is known. We consider a glueable select&verify graph problem $\mathcal{G}$ defined by $(P_{\cdot}, \text{opt})$. The evaluation of the term $\text{opt}_{\boldsymbol{x}\in\{0,1\}^n} P_G(\boldsymbol{x})$ (cf. Definition 3.1) can be done recursively according to the following strategy.

Algorithm 4.1 1. Start the computation by evaluating

$$\text{opt}_{\boldsymbol{x}\in\{0,1\}^n} P_G(\boldsymbol{x}) \;=\; \text{opt}_{\boldsymbol{x}\in\{0,1\}^n,\, \boldsymbol{x}\sim\chi_0^{\text{ext}}} P_{G^{\text{ver}}}(\boldsymbol{x} \mid G^{\text{sel}}),$$

where "$\chi_0^{\text{ext}} \equiv \#$" is the everywhere undefined external coloring and $G^{\text{ver}} = G^{\text{sel}} = G$ (also cf. Definition 3.4).

2. 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 subgraphs $G^{\text{sel}}, G^{\text{ver}} \subseteq G$, and an external coloring $\chi^{\text{ext}} : V(G) \to C_0^{\text{ext}} + C_1^{\text{ext}} + \{\#\}$, we do the following:
   (a) If G^{ver} has size greater than some constant c, then find a $\sqrt{\cdot}$-separator S for G^{ver} with $V(G^{\text{ver}}) = A_1 + S + A_2$.
   (b) Define $\Phi := \{\chi^{\text{int}} : S \to C_0^{\text{int}} + C_1^{\text{int}} \mid \chi^{\text{ext}} \rightsquigarrow \chi^{\text{int}}\}$.
   For all internal colorings $\chi^{\text{int}} \in \Phi$ do:
      i. Determine $\text{Eval}_{A_i}(\chi^{\text{int}})$ recursively for $i = 1, 2$.
      ii. Determine $\text{Eval}_S(\chi^{\text{int}})$.
   (c) Return $\text{opt}_{\chi^{\text{int}}\in\Phi}\, h(\text{Eval}_{A_1}(\chi^{\text{int}}), \text{Eval}_S(\chi^{\text{int}}), \text{Eval}_{A_2}(\chi^{\text{int}}))$.

The sizes of the subproblems, i.e., the sizes of the graphs $G_{A_i}^{\text{ver}}(\chi^{\text{int}})$ which are used in the recursion, play a crucial role in the analysis of the running time of this algorithm. A particularly nice situation is given by the following problems.

Definition 4.2. *A glueable select&verify problem is called* slim *if the subgraphs $G_{A_i}^{ver}(\chi^{int})$ are only by a constant number of vertices larger than $G^{ver}[A_i]$, i.e., if there exists an $\eta \geq 0$ such that $|V(G_{A_i}^{ver}(\chi^{int}))| \leq |A_i| + \eta$ for all internal colorings $\chi^{int} : S \to C^{int}$.*

Note that the proof of Lemma 3.9 shows that both Vertex Cover and Independent Set are slim with $\eta = 0$, whereas Dominating Set is not.

The following proposition gives the running time of the above algorithm in terms of the parameters of the separator theorem used and the select&verify problem considered. 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 "sufficiently small" separator in an n-vertex graph from class $\mathbb{G}$.
- $T_M(n)$ denotes the time to construct the modified graphs $G_X^{\text{ver}}(\chi^{\text{int}}) \in \mathbb{G}$ and the modified colorings $(\chi^{\text{ext}} \oplus_X \chi^{\text{int}})$ (for $X \in \{A_1, S, A_2\}$ and each internal coloring $\chi^{\text{int}} \in \Phi$ from an n-vertex graph from class $\mathbb{G}$.
- $T_E(m)$ is the time to evaluate $\text{Eval}_S(\chi^{\text{int}})$ for any $\chi^{\text{int}} \in \Phi$ in a separator S of size m.
- $T_G(n)$ 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 4.3. *Let $\mathbb{G}$ be a graph class for which a $\sqrt{\cdot}$-separator theorem with constants α and β is known and let $\mathcal{G}$ be a select&verify problem defined by $(P_\cdot, \text{opt})$ that is glueable with σ colors. Then, for every $G \in \mathbb{G}$, $\text{opt}_{\boldsymbol{x}\in\{0,1\}^n} P_G(\boldsymbol{x})$ 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, $\alpha' = \alpha + \varepsilon$ for any $\varepsilon \in (0, 1-\alpha)$ and $q(\cdot)$ is some polynomial; the running time analysis only holds for $n \geq n_0(\varepsilon)$.

If, however, $\mathcal{G}$ is slim or the $\sqrt{\cdot}$-separator theorem yields cycle separators (and $\mathbb{G}$ is the class of planar graphs), then the running time for the computation is $c(\alpha, \beta, \sigma)^{\sqrt{n}} q(n)$, which then holds for all n.

Proof. Let $T(n)$ denote the running time to compute

$$\text{opt}_{\boldsymbol{x}\in\{0,1\}^n,\ \boldsymbol{x}\sim\chi^{\text{ext}}} P_{G^{\text{ver}}}(\boldsymbol{x} \mid G^{\text{sel}})$$

for $G^{\text{ver}} = (V^{\text{ver}}, E^{\text{ver}})$ with $n = |V^{\text{ver}}|$ (where $\chi^{\text{ext}} : V(G) \to C_0^{\text{ext}} + C_1^{\text{ext}} + \{\#\}$ is some external coloring and $G^{\text{sel}}, G^{\text{ver}} \subseteq G$). In general, the recurrence we have to solve in order to compute an upper bound on $T(n)$ for the above given divide-and-conquer algorithm then reads as follows:

$$T(n) \leq \sigma^{\beta\sqrt{n}} \cdot 2\, T(\alpha n + \beta\sqrt{n}) \cdot T_{M,E,G}(n) + T_S(n), \quad \text{where}$$

$$T_{M,E,G}(n) := T_M(n) + T_E(\beta\sqrt{n}) + T_G(\alpha n + \beta\sqrt{n}).$$

By assumption, a separation (A_1, S, A_2) of G^{ver} can be found in time $T_S(n)$. Since the size of the separator is upperbounded by $\beta\sqrt{n}$ and there are σ internal colors, there are $\sigma^{\beta\sqrt{n}}$ many passes through a loop which checks all possible assignments to separator vertices. In each pass through the loop, two instances of smaller subproblems have to be evaluated, namely for A_1 and A_2. This yields the term $2 \cdot T(\alpha n + \beta\sqrt{n})$. The time needed for modifying the graphs and colorings, for evaluating the separator and for gluing the obtained subproblems is covered by $T_{M,E,G}(n)$. Note that the functions $T_{M,E,G}(n)$ and $T_S(n)$ are polynomials by our general assumption. From the definition of glueability, we have that the size of the two remaining subproblems to be solved recursively is

$$|V(G^{\mathrm{ver}}_{A_i}(\chi^{\mathrm{int}}))| \leq |V(G^{\mathrm{ver}}[\delta A_i])|\alpha n + \beta\sqrt{n}$$

for each $\chi^{\mathrm{int}} \in \Phi$. For every $\varepsilon \in (0, 1-\alpha)$, there is, of course, an $n_0(\varepsilon)$ such that

$$(\alpha + \varepsilon)n \geq \alpha n + \beta\sqrt{n} \tag{3}$$

for $n \geq n_0(\varepsilon)$. Hence, by setting $\alpha' = \alpha + \varepsilon$, we can simplify the above recurrence to

$$T(n) \leq 2 \cdot \sigma^{\beta\sqrt{n}}\, T(\alpha' n)\, T_{M,E,G}(n) + T_S(n)$$

for some $\alpha' < 1$. Hence, the recursion depth is $n' = \log_{1/\alpha'}(n)$, and we get

$$\begin{aligned} T(n) \leq\ & \sigma^{\beta \sum_{i=0}^{n'} \sqrt{\alpha'^i n}}\, 2^{n'} T(O(1)) \textstyle\prod_{i=0}^{n'} T_{M,E,G}(\alpha'^i n) + n' \cdot T_S(n) \\ \leq\ & \sigma^{\beta/(1-\sqrt{\alpha'})\cdot\sqrt{n}}\, q(n) \end{aligned}$$

for the polynomial $q(n) := 2^{\log_{1/\alpha'} n} T(O(1)) \prod_{i=0}^{n'} p(\alpha'^i n) + \log_{1/\alpha'}(n) \cdot T_S(n)$.

Now, consider the case where $\mathcal{G}$ is slim. In this situation the recursive subproblems have size $|V(G^{\mathrm{ver}}_{A_i}(\chi^{\mathrm{int}}))| \leq |A_i| + \eta$. Hence, we have

$$T(n) \leq \sigma^{\beta\sqrt{n}} \cdot 2\, T(\alpha n + \eta) \cdot T_{M,E,G}(n) + T_S(n).$$

The size of the problem after r recursion steps obviously is:

$$\alpha^r n + \alpha^{r-1}\eta + \cdots + \alpha\eta + \eta \leq \alpha^r n + \frac{1}{1-\alpha}\eta.$$

Since the recursion will stop after $n' = \log_{1/\alpha} n$ recursive calls (leaving us with subproblems of at most constant size), we can further estimate:

$$\begin{aligned} T(n) &\leq \sigma^{\beta \sum_{i=0}^{n'} \sqrt{\alpha^i n + \eta/(1-\alpha)}}\, 2^{n'} T(O(1)) \prod_{i=0}^{n'} T_{M,E,G}(\alpha^i n + \eta/(1-\alpha)) + n' \cdot T_S(n) \\ &\leq \sigma^{\beta\cdot(\sum_{i=0}^{n'} \sqrt{\alpha^i n} + (n'+1)\cdot\sqrt{\eta/(1-\alpha)})}\, 2^{n'} T(O(1)) \prod_{i=0}^{n'} T_{M,E,G}(\alpha^i n + \eta/(1-\alpha)) \\ &\qquad + n' \cdot T_S(n) \\ &\leq \sigma^{\beta/(1-\sqrt{\alpha})\cdot\sqrt{n}} q(n) \end{aligned}$$

for some polynomial $q(\cdot)$. More precisely, we can estimate

$$q(n) \leq \sigma^{\beta\sqrt{\eta/(1-\alpha)}(n'+1)} 2^{n'} T(O(1)) \prod_{i=0}^{n'} T_{M,E,G}(\alpha^i n + \eta/(1-\alpha)) + n' T_S(n)\,.$$

It remains to prove the claim in the case of the existence of a cycle separator theorem. Without going into detail, we want to sketch the key idea in this case. We consider the first recursive step of the algorithm, where we deal with a separation (A_1, S, A_2) of the input graph G. Suppose now (A_{11}, S_1, A_{12}) is a separation of $G[A_1]$, then, for $v \in S$, it is possible that $N(v) \cap A_{1i} \neq \emptyset$ for both $i = 1, 2$. This phenomenon basically forces us to find the separator in the first recursive step in $G_{A_1}(\chi^{\text{int}})$ (a possibly larger graph than $G[A_1]$) rather than in $G[A_1]$. In the case of cycle separators, i.e., if S_1 were a cycle separator, the above described phenomenon cannot occur anymore due to the Jordan curve theorem, also see Remark 2.5. That is why Algorithm 4.1 can be modified such that the new separator in step 2a is computed for G^{sel} (instead of G^{ver}). The corresponding recurrence equation for this modified algorithm then reads as

$$T(n) \leq \sigma^{\beta\sqrt{n}} \cdot 2\,T(\alpha n) \cdot T_{M,E,G}(n) + T_S(n),$$

the solution of which is given by $T(n) \leq \sigma^{\beta/(1-\sqrt{\alpha})\cdot\sqrt{n}} q(n)$ for some polynomial $q(\cdot)$. □

Remark 4.4. A similar proposition holds for graph classes on which an ℓ-separator theorem is known with constants α and β. It might turn out that such separator theorems have better ratio $\beta/(1-\sqrt{\alpha})$, which, in turn, would directly improve the running time in Proposition 4.3.

4.2 How (linear) problem kernels help

If the considered parameterized problem has a problem kernel of size dk, we can use the considerations we have made up to this point in order to obtain fixed-parameter algorithms whose exponential term is of the form $c^{\sqrt{k}}$ for some constant c. More generally, a problem kernel of size $p(k)$ yields exponential terms of the form $c^{\sqrt{p(k)}}$.

Theorem 4.5. *Assume the following:*

- *Let* $\mathbb{G}$ *be a graph class for which a* $\sqrt{\cdot}$*-separator theorem with constants* α *and* β *is known,*
- *let* $\mathcal{G}$ *be a select&verify problem defined by* $(P_\cdot, \text{opt})$ *glueable with* σ *colors, and*
- *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*

$$c(\alpha', \beta, \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k), \quad \textit{where} \quad c(\alpha', \beta, \sigma) = \sigma^{\beta/(1-\sqrt{\alpha'})}, \tag{4}$$

and $\alpha' = \alpha + \varepsilon$ *for any* $\varepsilon \in (0, 1-\alpha)$*, holding only for* $k \geq k_0(\varepsilon)$*, where* $q(\cdot)$ *is some polynomial.*

If, however, $\mathcal{G}$ is slim or the $\sqrt{\cdot}$-separator theorem yields cycle separators (on the class $\mathbb{G}$ of planar graphs), 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.

Proof. The result directly follows from Proposition 4.3 applied to the problem kernel. □

In particular, Theorem 4.5 means that, for glueable select&verify problems for planar graphs that admit a *linear* problem kernel of size dk, we can get an algorithm of running time

$$O\big(c(\alpha, \beta, \sigma, d)^{\sqrt{k}} q(k) + T_K(n, k)\big), \quad \text{where} \quad c(\alpha, \beta, \sigma, d) = \sigma^{\sqrt{d}\beta/(1-\sqrt{\alpha})}.$$

Obviously, the choice of the separator theorem has a decisive impact on the constants of the corresponding algorithms. In particular, our running time analysis shows that the ratio $r(\alpha, \beta) := \beta/(1 - \sqrt{\alpha})$ has a direct and significant influence on the running time. In Table 1, this ratio is computed for the various $\sqrt{\cdot}$-separator theorems. In the following example we use these ratios explicitly.

Example 4.6. In the case of Vertex Cover on planar graphs, we can take $d = 2$, $\alpha = 2/3$, and $\beta = \sqrt{2/3} = \sqrt{4/3}$ (see [17]) with the ratio $r(\alpha, \beta) \approx 10.74$. In this way, we obtain an algorithm with running time $O(2^{\sqrt{2}\cdot 10.74 \cdot \sqrt{k}} + nk)$. Neglecting polynomial terms, we have such obtained a $c^{\sqrt{k}}$-algorithm with $c \approx 2^{15.19} \approx 37381$. By way of contrast, taking $d = 2$, $\alpha = 3/4$, and $\beta = \sqrt{2\pi/\sqrt{3}} \cdot (1 + \sqrt{3})/\sqrt{8} \approx 1.84$ (see [32]) with $r(\alpha, \beta) \approx 13.73$, we get an algorithm with running time $O(2^{\sqrt{2}\cdot 13.73 \cdot \sqrt{k}} + nk)$. This means, we have a $c^{\sqrt{k}}$-algorithm with $c \approx 2^{19.42} \approx 701459$.

The constants obtained by this first approach are admittedly bad. Section 5 is dedicated to present new strategies on how to substantially improve these constants.

Remark 4.7. We discuss the importance of cycle separator theorems or slim graph problems. Assume that none of these two conditions is met in a given situation. Then, the claimed bound from Equation (4) of Theorem 4.5 is only true for some $\alpha' = \alpha + \varepsilon$ with $\varepsilon \in (0, 1 - \alpha)$. Now, there is a certain trade-off in the choice of ε:

1. The factor $\beta/(1 - \sqrt{\alpha'})$ in the exponent of $c(\alpha', \beta, \sigma)$ tends to infinity if α' tends to one, i.e., if ε is as large as possible.
2. The analysis of Theorem 4.5 is only valid if $p(k) \geq (\beta/\varepsilon)^2$. This bound is easily derived from Equation (3) in Proposition 4.3 when replacing n by $p(k)$ due to the assumption of a problem kernel of polynomial size.

Keeping in mind that typical values of $p(k)$ are not very large in practical cases, the second point means that, since β is fixed, ε should be comparatively large, otherwise, β/ε would be greater than $\sqrt{p(k)}$. This gives us very bad constants in the analysis due to the first point.

As explained in the following example, Theorem 4.5 is not only interesting in the case of planar graph problems.

Example 4.8. Since Vertex Cover is a slim problem which has a linear size kernel, Theorem 4.5 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 [16], where the existence of a separator of size $O(\sqrt{gn})$ for n-vertex graphs from $\mathbb{G}_g$ was proven. For the same reason, we get a $c^{\sqrt{gk}}$-algorithm for Independent Set on $\mathbb{G}_g$. Similarly, for the class of ℓ-map graphs, we obtain $c^{\sqrt{\ell k}}$-algorithms for Vertex Cover and for Independent Set based on [13].

Note that these are the first examples of fixed-parameter algorithms with sublinear exponents for bounded genus graphs and for ℓ-map graphs. In this sense, the techniques discussed in this paper apply to a wider range of graphs compared to the approach that in [5]. The "Layerwise Separation Property" there makes sense for planar graphs only, although it might be extensible to map graphs.

5 Improving constants

We present strategies to lower the admittedly bad base of the exponential term obtained by the worst-case analysis in Theorem 4.5. On the one hand, we obtain direct improvements by a closer analysis of the separator theorems (see Subsection 5.1). On the other hand, allowing the sublinear exponent to be $O(k^{2/3})$ instead of $O(k^{1/2})$, we can further bring down the exponential base (see Subsection 5.2).

5.1 Analyzing separator theorems

We briefly analyze how (planar) separator theorems are proven in the literature. For the ease of presentation, we will again restrict ourselves to planar graphs. Similar observations can be made for other graph classes, as well.

A small separator, as obtained by Lipton and Tarjan's approach, consists of two ingredients coming from two separate steps of the construction:

Folding step: the first ingredient of the separator is composed of all vertices which have the same distance from the root of an assumed minimal height spanning tree (modulo some constant s), and

Bounded radius separation: the second ingredient is found according to a "special separator theorem" for planar graphs with radius of at most s, which guarantees the existence (and construction) of a separator containing no more than $2s+1$ vertices.

The constant s is chosen such that the overall size of the separator is minimized.

When making use of separator theorems in a recursive fashion, it is not necessary to perform a folding step in each recursion step, since the bounded radius assumption of the special separator theorem is still met. Only from time to time, it is necessary to cut down the radius s of the remaining graph parts by interleaving a folding step.

Following the described venue, we were able to prove the following theorem whose detailed proof can be found in [4].

Theorem 5.1. *Let $\mathcal{G}$ be a select&verify problem on planar graphs defined by $(P_\cdot, \mathrm{opt})$ which is glueable with σ colors, and suppose that $\mathcal{G}$ admits a problem kernel of polynomial size $p(k)$ computable in time $T_K(n,k)$.*

Then, there is an algorithm to decide $(G,k) \in \mathcal{G}$, for an n-vertex planar graph G, in time

$$c(\alpha', \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k), \quad \textit{where} \quad c(\alpha', \sigma) \approx \sigma^{1.80665/(1-\sqrt{\alpha'})},$$

and $\alpha' = 2/3 + \varepsilon$ for any $\varepsilon \in (0, 1/3)$, holding only for $k \geq k_0(\varepsilon)$, where $q(\cdot)$ is some polynomial.

If $\mathcal{G}$ is slim, then the running time for the computation is

$$c(2/3, \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k),$$

which then holds for all k. □

Example 5.2. For Vertex Cover on planar graphs, we obtain by the previous theorem a $2^{13.9234\sqrt{k}} \approx 15537^{\sqrt{k}}$-algorithm, which obviously beats the figures in Example 4.6.

To further improve on our constants, it is possible to analyze a planar $\sqrt{\cdot}$-separator theorem yielding 3-separators (due to Djidjev [15]) in essentially the same way as sketched above, leading to the following theorem:

Theorem 5.3. *Let $\mathcal{G}$ be a select&verify problem on planar graphs defined by $(P_\cdot, \mathrm{opt})$ which is 3-glueable with σ colors, and suppose that $\mathcal{G}$ admits a problem kernel of polynomial size $p(k)$ computable in time $T_K(n,k)$.*

Then, there is an algorithm to decide $(G,k) \in \mathcal{G}$, for an n-vertex planar graph G, in time

$$c(\alpha', \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k), \quad \textit{where} \quad c(\alpha', \sigma) \approx \sigma^{2.7056/(1-\sqrt{\alpha'})},$$

and $\alpha' = 1/2 + \varepsilon$ for any $\varepsilon \in (0, 1/2)$, holding only for $k \geq k_0(\varepsilon)$, where $q(\cdot)$ is some polynomial.

If $\mathcal{G}$ is slim, then the running time for the computation is

$$c(1/2, \sigma)^{\sqrt{p(k)}} q(k) + T_K(n,k),$$

which then holds for all k. □

Example 5.4. For Vertex Cover on planar graphs, we obtain in this way a $2^{13.0639\sqrt{k}} \approx 8564^{\sqrt{k}}$-algorithm, which again beats the figure derived in Example 5.2.

Observe that Theorem 5.3 is not always yielding a better algorithm than Theorem 5.1, since possibly more colors are needed in the recursion for 3-glueability than for (2-)glueability, see Remark 3.11.

Remark 5.5. In the case that a slim select&verify planar graph problem $\mathcal{G}$, which is glueable with σ colors, admits a kernel of size dk, where the kernelization yields a problem parameter $k' = k$ (for simplicity), two principle methods are immediately at hand:

- Test all possible 0-1-settings in order to obtain a 2^{dk}-algorithm for $\mathcal{G}$.
- Employ Theorem 5.3 in order to get a $\sigma^{2.7056/(1-\sqrt{1/2})\sqrt{dk}} = \sigma^{9.2376\sqrt{dk}}$-algorithm.

Taking, e.g., the 2^{dk}-algorithm for $\mathcal{G}$ and equating 2^{dk} with $\sigma^{9.2376\sqrt{dk}}$ yields

$$k_{BE}(\sigma, d) = \left(\frac{9.2376 \cdot \log(\sigma)}{\log(2)}\right)^2 \frac{1}{d} = 177.61 \cdot (\log(\sigma))^2 d^{-1} \qquad (5)$$

as *break even point* of the separator-based algorithm, i.e., whenever $k \geq k_{BE}(\sigma, d)$, then the separator-based algorithm is better than the 2^{dk}-algorithm.

In many cases, as for example in the case of Vertex Cover, there are known so-called search tree algorithms which are much better than the 2^{dk}-algorithm. Therefore, the break even point of the separator-based algorithm tends to be greater than suggested by Equation (5). For example, in the case of Vertex Cover, a 1.3^k-algorithm beats the suggested $8564^{\sqrt{k}}$-algorithm as long as $k \leq 1191$. Of course, $k \approx 1000$ is out of reach of current computer technology, although one has to keep in mind that we are always talking about worst-case upper bounds. For example, the pieces obtained by the separator theorems might be much smaller than suggested by the theorems, which immediately yields an improved running time of these algorithms. Nevertheless, in practice, it might be favorable to devise algorithms which have worse asymptotic behavior (since the exponential term contains a faster growing function) but smaller constants. We try to capture this sort of trade-off in the following.

5.2 n/ε separation

There are separator theorems which allow for arbitrary small weights (upper-bounded by ε) for each of the (many) graph components into which the given graph is split, at the expense of getting larger separators (of a size depending also on the chosen ε). Aleksandrov and Djidjev proved the following result, which is, to our knowledge, the best of its kind [7]:

Lemma 5.6. *Let G be an n-vertex planar graph with nonnegative vertex weights adding to at most one, and let $0 < \varepsilon \leq 1$. Then, there exists some separator S with at most $4\sqrt{n/\varepsilon}$ many vertices such that $G - S$ has no connected component of total weight greater than ε. The separator S can be found in $O(n)$ time.* □

Let us briefly sketch the proof of Lemma 5.6: As in the proofs of previously shown separator theorems, in a preparatory folding step, the graph is sliced into pieces of bounded radius s by cutting off a layer L_{T,i_0} with $|L_{T,i_0}| \leq n/s$. Then, the special separator theorem for bounded radius graphs is used repeatedly, until all remaining graph pieces are small enough. Choosing an optimal s then yields the claimed constant.

We want to apply Lemma 5.6 in order to design algorithms for certain select&verify planar graph problems that are glueable with σ colors. Thinking about ε as a function $\varepsilon : n \mapsto (0, 1]$ allows us to devise an algorithm of the following kind:

- Apply Algorithm 4.1, as long some of the graph pieces such obtained in the course of the recursion have more than $\varepsilon(n)n$ many vertices, and
- compute an optimal solution by using the best-known $\rho^{\varepsilon(n)n}$ algorithm (for the precolored problem!) on each graph piece. Of course, there are at most n such graph pieces.

Let us try to determine an optimal $\varepsilon(n)$, assuming that the constant ρ is known. Recall that only the size of the separators on some path of the divide-and-conquer recursion tree matters for the calculation of the running time. This size is upperbounded by

$$n/s + 2s \log_{3/2}(1/\varepsilon(n)), \tag{6}$$

since the depth of the recursion tree is bounded by $\log_{3/2}(1/\varepsilon(n))$. Expression (6) is minimal if $n/s = 2s \log_{3/2}(1/\varepsilon(n))$. This means that choosing

$$s(n) = \sqrt{\frac{\log(3/2)}{2}} \sqrt{\frac{n}{-\log(\varepsilon(n))}} \tag{7}$$

would be optimal. Now, the task would be to determine the optimal function $\varepsilon(n)$ in order to minimize the total running time, which is upperbounded by

$$(\sigma^{s(n)} \cdot \rho^{\varepsilon(n)n})n. \tag{8}$$

For fixed n, the minimum of Expression (8) is assumed when

$$\sigma^{s(n)} = \rho^{\varepsilon(n)n}.$$

Unfortunately, we were not able to determine the optimal function $\varepsilon(n)$ corresponding to the $s(n)$ given by Equation (7) analytically. Therefore, we make the following coarse estimate: Of course, the size of the separators along one recursion path is upperbounded by the size of all separators which is upperbounded by $4\sqrt{n/\varepsilon}$ according to Lemma 5.6. Therefore, we compute the $\varepsilon(n)$ given by

$$\sigma^{4\sqrt{n/\varepsilon(n)}} = \rho^{\varepsilon(n)n}. \tag{9}$$

This means that $\varepsilon(n) = \Theta(n^{-1/3})$ is the best choice. More precisely, evaluating Equation (9) yields

$$\varepsilon(n) = \left(\frac{\log(\sigma) \cdot 4}{\log(\rho)}\right)^{2/3} n^{-1/3}. \tag{10}$$

Hence, an algorithm which tries all possible colorings along a recursion path, applying the best known general algorithm to the remaining graph pieces, results in the following running time due to Expression (8):

$$(\sigma^{s(n)} \cdot \rho^{\varepsilon(n)n})n,$$

where we use $s(n) = 4\sqrt{n/\varepsilon(n)}$ and $\varepsilon(n)$ is given by Equation (10).

This reasoning yields the following statement.

Theorem 5.7. *Let $\mathcal{G}$ be a select&verify problem on planar graphs, defined by $(P_\cdot, \mathrm{opt})$. We make the following further assumptions:*

- *$\mathcal{G}$ is glueable with σ colors.*
- *$\mathcal{G}$ admits a problem kernel of linear size dk computable in time $T_K(n,k)$.*
- *There is an $O(\rho^n)$ algorithm for solving the (possibly precolored) graph decision problem under consideration for a planar graph with n vertices.*

Then, there is an algorithm to decide $(G,k) \in \mathcal{G}$, for an n-vertex planar graph G, in time

$$O(dk \cdot 2^{\theta(\sigma,\rho,d)\cdot k^{2/3}} + T_K(n,k)), \quad \textit{where} \quad \theta(\sigma,\rho,d) = 2\log(\rho)\left(4d\frac{\log(\sigma)}{\log(\rho)}\right)^{2/3}.$$

□

Observe that we indeed need no further assumptions, since we rely on the special cycle separator theorem for bounded radius graphs from the second step in the recursion on. The first step of the recursion does not cause any problems.

In the case of Vertex Cover, Robson's algorithm [31] gives $\rho \approx 1.21$, so that Theorem 5.7 yields a $c^{k^{2/3}}$-algorithm with $c \approx 2^{(2\cdot\log 1.21\cdot(8/\log 1.21)^{2/3})} \approx 2^{5.20} \approx 36.81$. Recently announced improvements of [31][10] suggest taking $\rho \approx 1.18$, which would yield $c \approx 2^{4.96} \approx 31.19$. Compare this running time with the best known $c^{k^{1/2}}$-algorithm ($c = 2^{4\sqrt{3}} \approx 121.79$ was shown in [5]).

Similar remarks can be made for Independent Set. Here, in comparison with Vertex Cover, the problem kernel size is by a factor of two larger and, hence, the exponent gets worse by a factor of $2^{2/3} \approx 1.59$, yielding a $(2^{5.20\cdot 1.59})^{k^{2/3}} \approx (306.20)^{k^{2/3}}$-algorithm.

6 Conclusion

We exhibited the relations between (planar) separator theorems and their use in obtaining fixed-parameter tractability results based on divide-and-conquer algorithms. To this end, we coined the key notion of glueable select&verify problems

[10] Personal communication of J. M. Robson

that captures intricate graph problems such as DOMINATING SET or TOTAL DOMINATING SET. We showed that various glueable select&verify problems allow $c^{\sqrt{k}}$-algorithms on graph classes that admit a $\sqrt{\cdot}$-separator theorem. Then, the constant c is determined in terms of some problem-specific parameters. By exploiting further ideas on the use of separator theorems, we were able to lower these constants substantially. Finally, methods were presented that allow for $c^{k^{2/3}}$-algorithms with already reasonable constants c.

Our work indicates that research on improving constants in separator theorems with constants α and β should not only concentrate on bringing down β for fixed α (as, e.g., done for $\alpha = 2/3$ in [17]), but more importantly, on minimizing the function $\beta/(1-\sqrt{\alpha})$. This would directly improve the exponential terms in all graph problems captured by our methodology. Moreover, it is interesting to see whether the constants in Theorems 5.1 and 5.3 can be further improved. This should be possible, since we only analyzed the first ever proven separator theorems and were able to bring down the exponential term by a factor of 2/3. The idea behind our two theorems might also help to overcome the "lower bound barrier" imposed on the constants of several separator theorems, see [32]. Of course, an improvement of the presented algorithms can also be gained by increasing the number of parameterized problems with (small) linear problem kernel.

Last but not least, our techniques might be applicable to non-planar graphs, as well. This is strongly indicated by [8, 9, 19, 22], as well as by several remarks throughout this paper.

Acknowledgment We thank the anonymous referees of the 7th Annual International Computing and Combinatorics Conference (COCOON 2001) and of *Journal of Computer and System Sciences* for their remarks that helped improving the presentation of this paper.

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 Scandinavian Workshop on Algorithm Theory SWAT'00*, volume 1851 of *LNCS*, pages 97–110. Springer-Verlag, 2000. Long version to appear in *Algorithmica*.
2. J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege. Refined search tree techniques for the PLANAR DOMINATING SET problem. In *Proc. 26th International Symposium Mathematical Foundations of Computer Science MFCS'01*, volume 2136 of *LNCS*, pages 111–122. Springer-Verlag, 2001.
3. J. Alber, M. R. Fellows, and R. Niedermeier. Efficient data reduction for DOMINATING SET: a linear problem kernel for the planar case. To appear in *Proc. 8th Scandinavian Workshop on Algorithm Theory SWAT'02*, July 2002.
4. J. Alber, H. Fernau, and R. Niedermeier. Graph separators: a parameterized view. Technical Report WSI–2001–8, Universität Tübingen (Germany), Wilhelm-Schickard-Institut für Informatik, September 2001.

5. 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 International Colloquium on Automata, Languages and Programming ICALP'01*, volume 2076 of *LNCS*, pages 261–272. Springer-Verlag, 2001.
6. J. Alber, J. Gramm, and R. Niedermeier. Faster exact algorithms for hard problems: A parameterized point of view. *Discrete Mathematics*, 229:3–27, 2001.
7. L. G. Aleksandrov and H. N. Djidjev. Linear algorithms for partitioning embedded graphs of bounded genus. *SIAM Journal of Discrete Mathematics*, 9:129–150, 1996.
8. N. Alon, P. D. Seymour, and R. Thomas. A separator theorem for graphs with an excluded minor and its applications. In *Proc. 22nd ACM Symposium on the Theory of Computing STOC'90*, pages 293–299, 1990.
9. N. Alon, P. D. Seymour, and R. Thomas. A separator theorem for nonplanar graphs. *Journal of the AMS*, 3:801–808, 1990.
10. R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematcs*, 25:27–46, 1985.
11. H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.
12. J. Chen, I. Kanj, and W. Jia. Vertex cover: further observations and further improvements. In *Journal of Algorithms*, 41:280–301, 2001.
13. Z.-Z. Chen. Approximation algorithms for independent sets in map graphs. *Journal of Algorithms*, 41:20–40, 2001.
14. Z.-Z. Chen, M. Grigni, and C. H. Papadimitriou. Map graphs. To appear in *Journal of the ACM*, 2002.
15. H. N. Djidjev. On the problem of partitioning planar graphs. *SIAM J. Algebraic Discrete Methods*, 3(2):229–240, 1982.
16. H. N. Djidjev. A separator theorem for graphs of fixed genus. *Serdica*, 11:319–329, 1985.
17. H. N. Djidjev and S. M. Venkatesan. Reduced constants for simple cycle graph separation. *Acta Informatica*, 34:231–243, 1997.
18. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
19. D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27:275–291, 2000.
20. M. R. Fellows. Parameterized complexity: the main ideas and some research frontiers. In *Proc. 12th Annual International Symposium on Algorithms And Computation ISAAC'01*, volume 2223 of *LNCS*, pages 291–307, 2001.
21. M. R. Garey and D. S. Johnson. *Computers and Intractability.* New York: Freeman, 1979.
22. M. Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, To appear, 2001.
23. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal of Applied Mathematics*, 36(2):177–189, 1979.
24. R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.
25. G. L. Nemhauser and J. L. E. Trotter. Vertex packing: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
26. R. Niedermeier and P. Rossmanith. Upper bounds for Vertex Cover further improved. In *Proc. 16th Symposium on Theoretical Aspects of Computer Science STACS'99*, volume 1563 of *LNCS*, pages 561–570. Springer-Verlag, 1999.

27. R. Niedermeier and P. Rossmanith. On efficient fixed parameter algorithms for WEIGHTED VERTEX COVER. In *Proc. 11th Annual International Symposium on Algorithms And Computation ISAAC'00*, volume 1969 of *LNCS*, pages 180–191. Springer-Verlag, 2000.
28. V. T. Paschos. A survey of approximately optimal solutions to some covering and packing problems. *ACM Computing Surveys*, 29(2):171–209, 1997.
29. S. S. Ravi and H. B. Hunt. An application of the planar separator theorem to counting problems. *Information Processing Letters*, 25(6):317–322, 1987.
30. N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas. Efficiently four-coloring planar graphs. In *Proc. 28th ACM Symposium on the Theory of Computing STOC'96*, pages 571–575, 1996.
31. J. M. Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.
32. D. A. Spielman and S.-H. Teng. Disk packings and planar separators. In *SCG 96: 12th Annual ACM Symposium on Computational Geometry*, pages 349–358, 1996.
33. J. A. Telle. Complexity of domination-type problems in graphs. *Nordic Journal of Comp.*, 1:157–171, 1994.
34. J. A. Telle and A. Proskurowski. Practical algorithms on partial k-trees with an application to domination-like problems. In *Proc. 3rd Algorithms and Data Structures WADS'93*, volume 709 of *LNCS*, pages 610–621. Springer-Verlag, 1993.
35. J. A. Telle and A. Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM Journal of Discrete Mathematics*, 10(4):529–550, 1997.
36. M. Yannakakis and F. Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.