21st International Symposium on Theoretical Aspects of Computer Science, STACS 2004, Springer Verlag, Vol. 2996, pp. 56 – 67, 2004.

A Simple and Fast Approach for Solving Problems on Planar Graphs*

Fedor V. Fomin[1] and Dimitrios M. Thilikos[2]

[1] Department of Informatics, University of Bergen, N-5020 Bergen, Norway, fomin@ii.uib.no

[2] Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord – Mòdul C5, c/Jordi Girona Salgado 1-3, E-08034, Barcelona, Spain, sedthilk@lsi.upc.es

Abstract. It is well known that the celebrated Lipton-Tarjan planar separation theorem, in a combination with a divide-and-conquer strategy leads to many complexity results for planar graph problems. For example, by using this approach, many planar graph problems can be solved in time $2^{O(\sqrt{n})}$, where n is the number of vertices. However, the constants hidden in big-Oh, usually are too large to claim the algorithms to be practical even on graphs of moderate size. This paper aims to overcome this problem by introducing a new algorithm design paradigm for solving problems on planar graphs. The paradigm is so simple that it can be explained in any textbook on graph algorithms: Compute tree or branch decomposition of a planar graph and do dynamic programming. Surprisingly such a simple approach provides faster algorithms for many problems. For example, Independent Set on planar graphs can be solved in time $O(2^{3.182\sqrt{n}}n+n^4)$ and Dominating Set in time $O(2^{5.043\sqrt{n}}n+n^4)$. Moreover, a significantly broader class of problems can be attacked by this method. Thus with our approach, Longest cycle on planar graphs is solvable in time $O(2^{2.29\sqrt{n}(\ln n+0.94)}n^{5/4}+n^4)$ and Bisection is solvable in time $O(2^{3.182\sqrt{n}}n+n^4)$. The proof of these results is motivated by a recent combinatorial result stating that branch-width of a planar graph is at most $2.122\sqrt{n}$. In addition, we observe how a similar approach can be used for solving different fixed parameter problems on planar graphs. We prove that our method provides the best so far exponential speed-up for fundamental problems on planar graphs like Vertex Cover, (Weighted) Dominating Set, and many others.

1 Introduction

The design of (exponential) algorithms that are significantly faster than exhaustive search is one of the basic approaches of coping with NP-hardness [18]. Nice examples of fast exponential algorithms are Eppstein's graph coloring algorithm [17] and the algorithm for 3-SAT [12]. For a good overview of the field see the recent survey written by Gerhard Woeginger [34].

* This work was partially supported by the IST Program of the EU under contract number IST-1999-14186 (ALCOM-FT). The last author was also supported by the EU within the 6th Framework Programme under contract 001907 (DELIS) and by the Spanish CICYT project TIC-2002-04498-C05-03 (TRACER).

It is well known that by making use of the well-known approach of Lipton & Tarjan, [27] based on the celebrated planar separator theorem [26], one can obtain algorithms with time complexity $c^{O(\sqrt{n})}$ for many problems on planar graphs. However, the constants "hidden" in $O(\sqrt{n})$ can be crucial for practical implementations. During the last few years a lot of work has been done to compute and to improve the "hidden" constants [3, 4]. In this paper we propose a general approach for obtaining sub-exponential time *exact* algorithms for many problems on planar graphs. Our approach is based on dynamic programming for graphs of bounded branch-width (tree-width). Combining the best, so far, upper bound for the branch-width of planar graphs with this simple approach, one can obtain exponential speed-up for many known algorithms for many different planar graph problems. Independent Set, Dominating Set, MIN-Bisection, Longest Cycle (Path) on planar graphs are just a few examples of such problems.

Another field for taking advantage of the current bounds on treewidth and branch-width of planar graphs is the design of parameterized algorithms. The last ten years were the evidence of rapid development of a new branch of computational complexity: Parameterized Complexity. (See the book of Downey & Fellows [16].) Roughly speaking, a parameterized problem with parameter k is *fixed parameter tractable* if it admits a solving algorithm with running time $f(k)|I|^{\beta}$. (Here f is a function depending only on k, $|I|$ is the length of the non parameterized part of the input and β is a constant.) Typically, $f(k) = c^k$ is an exponential function for some constant k. During the last two years much attention was paid to the construction of parameterized algorithms with running time where $f(k) = c^{\sqrt{k}}$ for different problems on planar graphs. The first paper on the subject was the paper by Alber et al. [1] describing an algorithm with running time $O(4^{6\sqrt{34k}}n)$ (which is approximately $O(2^{70\sqrt{k}}n)$) for the Planar Dominating Set problem. Different fixed parameter algorithms for solving problems on planar and related graphs are discussed in [4, 25]. We observe that our technique can serve also as a simple unified approach for solving many parameterized problems on planar graphs in subexponential time. Again, our approach is based on combinatorial bounds on planar branch-width and treewidth and provides a better running time for such basic parameterized problems like Vertex Cover, Dominating Set and many others.

The aim of this paper is to show that such a simple approach, combined with the recent upper bound on the branch-width of planar graphs, *guarantees* better time bounds. More precisely, we use the recent upper bounds to the branch-width and the tree-width of planar graphs proved in [19]. Both these parameters where introduced (and served) as basic tools by Robertson and Seymour in their Graph Minors series of papers. Tree-width and branch-width are related parameters (See Theorem 1) and can be considered as measures of the "global connectivity" of a graph. Moreover, they appear to be of a major importance in algorithmic design as many NP-hard problems admit polynomial or even linear time solutions when their inputs are restricted to graphs of bounded tree-width or branch-width. This motivated the search for graphs where these parameters are relatively small. In this direction, Alon, Seymour & Thomas proved in [6] that given a minor closed graph class $\mathcal{G}$, any n-vertex graph G in $\mathcal{G}$ has tree-width/branch-width $O(\sqrt{n})$. As a consequence of this, any n-vertex planar graph G has tree-width/branch-width $\leq 14.697\sqrt{n}$.

In [19], it is shown that every n-vertex planar graph G has branch-width $\leq 2.122\sqrt{n}$ and tree-width $\leq 3.182\sqrt{n}$. To our knowledge, this is the best known upper bound for the value of these parameters on planar graphs. Any possible improvement of this bound will imply further acceleration of the algorithms described in this paper.

The paper is organised as follows: In Section 2, we give an overview of method and our results and provide some comparison with previous ones. In Section 3, we give some basic definitions and present the combinatorial bounds on treewidth and branchwidth, while in Section 4, we discuss their applications on the design of fast (subexponential) exact and parameterized algortihms.

2 Previous results and our contribution

Lipton & Tarjan [27] were first to observe the existence of time $2^{O(\sqrt{n})}n^{O(1)}$ algorithms for several problems on planar graphs. However the constants hidden in big-Oh of the exponent make these algorithms unpractical. Later, a lot of work was done on computing and reducing these constants. The best known so far results can be found in [4], where generalizations and complicated improvement of Lipton-Tarjan (together with kernel reduction techniques) are used to obtain subexponential parameterized algorithms.

Thus, for example, the approach suggested in [4] provides an $O(2^{9.07\sqrt{n}}n\ln n)$ algorithm for Independent Set and an $O(2^{18.61\sqrt{n}}n\ln n)$ algorithm for Dominating Set.

Here we suggest a unified approach based on branch decompositions (see Section 3 for the definitions). Our algorithm is simple and is performed in two steps: First we compute the branch decomposition of the planar graph of the input and then do dynamic programming on graphs of bounded branch-width. Optimal branch decomposition of a planar graph can be constructed in polynomial time by using the algorithm due to Seymour & Thomas (Sections 7 and 9 in [32]). (See also the results of Hicks [23] on implementations of Seymour & Thomas algorithm.) For graphs with n vertices this algorithm can be implemented in $O(n^4)$ steps. And what is important for practical applications, there is no *large hidden constants* in the running time of this algorithm. As for the second stage, well known dynamic programming algorithms on tree decompositions can be easily translated to branch decompositions. Using the current upper bounds for branch-width we prove that our approach provides *more efficient* solutions for many well known problems on planar graphs.

The following table summarize some known and new results on some problems on planar graphs (for more problems see Section 4).

	Known results	New results
Planar Independent Set	$O(2^{9.07\sqrt{n}} n \ln n)$ [4]	$O(2^{3.182\sqrt{n}} n + n^4)$
Planar Dominating Set	$O(2^{18.61\sqrt{n}} n \ln n)$ [4]	$O(2^{5.043\sqrt{n}} n + n^4)$
Planar (k,r)-center		$O((2r+1)^{3.182\sqrt{n}} n + n^4)$
Planar Longest Cycle		$O(2^{4.58\sqrt{n}(\frac{1}{2}\ln\sqrt{n}+0.47)} n^{5/4} + n^4)$
Planar Longest Path		$O(2^{4.58\sqrt{n}(\frac{1}{2}\ln\sqrt{n}+0.47)} n^{5/4} + n^4)$
Planar Bisection		$O(2^{3.182\sqrt{n}} n + n^4)$
Planar Weighted Dominating Set		$O(2^{6.37\sqrt{n}} n + n^4)$
Planar Perfect Code		$O(2^{6.37\sqrt{n}} n + n^4)$
Planar Red Blue Dominating Set		$O(2^{6.37\sqrt{n}} n + n^4)$
Planar H-coloring		$O(2^{\log h \cdot 2.12\sqrt{n}} h n^{3/2} + n^4)$
Planar Kernel		$O(2^{3.37\sqrt{n}} n^2 + n^4)$
Planar H-covering		$O(2^{9.55\sqrt{n}h} n + n^4)$

Similar approach works well also for parameterized problems. The next table summarizes results on the most fundamental fixed parameter problems on planar graphs. (See [3] for an overview of the results on this subject.) We include the result from [20] because it is based on the main combinatorial result of this paper and is obtained by similar approach.

	Known results	New results
Planar k-Vertex Cover	$O(2^{4\sqrt{3k}} n)$ [3]	$O(2^{4.5\sqrt{k}} k + k^4 + kn)$
Planar k-Dominating Set	$O(2^{27\sqrt{k}} n)$ [25]	$O(2^{15.13\sqrt{k}} k + k^4 + n^3)$[20]
Planar k-Independent Set	$O(2^{4\sqrt{6k}} n)$ [3]	$O(k^4 + 2^{4\sqrt{4.5k}} k + n)$

Thus our approach provides exponential speedup for the main basic parameterized problems. Our method is quite universal and can be implemented to obtain an exponential speed-up for many known algorithms for different problems with fixed parameters. Mention just a few parameterized versions of the following problems: Independent Dominating Set, Perfect Dominating Set, Perfect Code, Weighted Dominating Set, Total Dominating Set, Edge Dominating Set, Face Cover, Vertex Feedback Set, Minimum Maximal Matching, Clique Transversal Set, Disjoint Cycles, and Digraph Kernel. Another advantage of our results is that they apply not only on planar graphs but on different generalizations of planar graphs, e.g. $K_{3,3}$-minor-free or K_5-minor-free graphs.

3 Definitions and preliminary results

All graphs in this paper are undirected, loop-less and, unless otherwise mentioned, they may have multiple edges.

3.1 Tree-width

The notion of treewidth was introduced in [28] by Roberton and Seymour and, along with branchwidth, served as a basic tool in their Graph Minors series.

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

(1) $\bigcup_{i \in V(T)} X_i = V(G)$,
(2) for each edge $\{v, w\} \in E(G)$, there is an $i \in V(T)$ such that $v, w \in X_i$, and
(3) for each $v \in V(G)$ the set of nodes $\{i \mid v \in X_i\}$ forms a subtree of T.

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

3.2 Branch-width

Branchwidth was introduced in [29] as a parameter alternative to treewidth.

A *branch decomposition* of a graph (or a hyper-graph) G is a pair (T, τ), where T is a tree with vertices of degree 1 or 3 and τ is a bijection from the set of leaves of T to $E(G)$. The *order* of an edge e in T is the number of vertices $v \in V(G)$ such that there are leaves t_1, t_2 in T in different components of $T(V(T), E(T) - e)$ with $\tau(t_1)$ and $\tau(t_2)$ both containing v as an endpoint.

The *width* of (T, τ) is the maximum order over all edges of T, and the *branch-width* of G, $\mathbf{bw}(G)$, is the minimum width over all branch decompositions of G. (In case where $|E(G)| \leq 1$, we define the branch-width to be 0; if $|E(G)| = 0$, then G has no branch decomposition; if $|E(G)| = 1$, then G has a branch decomposition consisting of a tree with one vertex – the width of this branch decomposition is considered to be 0).

It is easy to see that if H is a subgraph of G then $\mathbf{bw}(H) \leq \mathbf{bw}(G)$. The following result is due to Robertson & Seymour [(5.1) in [29]].

Theorem 1 ([29]). *For any connected graph G where $|E(G)| \geq 3$, $\mathbf{bw}(G) \leq \mathbf{tw}(G) + 1 \leq \frac{3}{2}\mathbf{bw}(G)$.*

From Theorem 1, any upper bound on tree-width implies an upper bound on branch-width and vice versa. The proof of the following theorem is long and complicated. Currently it is available as technical report [19] and we shall publish it separately in a mathematical journal. The proof of the theorem makes strong use of deep "dual" and "min-max" theorems from Graph Minors series papers of Robertson & Seymour, in particular it is based on the relations between slopes and majorities defined in [32, 30] and [7] respectively.

Theorem 2. *For any planar graph G, $\mathbf{bw}(G) \leq \sqrt{4.5|V(G)|} \leq 2.122\sqrt{|V(G)|}$.*

Corollary 1. *For any planar graph G, $\mathbf{tw}(G) \leq \frac{3}{2}\sqrt{4.5|V(G)|} \leq 3.182\sqrt{|V(G)|}$.*

4 Algorithmic consequences

In this section we discuss the applications of Theorem 2 and Collorary 1 for different problems on planar graphs.

4.1 Subexponential exact algorithms

The following simple theorem is the source for obtaining subexponential algorithms for many graph problems.

Theorem 3. *Let Π be an optimization problem that is solvable on graphs of branch-width $\leq \ell$ in time $f(\ell)g(n)$, provided that a branch decomposition of width at most ℓ is given. Then on planar graphs problem Π is solvable in time $O(f(2.122\sqrt{n})g(n) + n^4)$*

Proof. First we compute an optimal branch decomposition of planar graph. To compute an optimal branch decomposition of a planar graph one can use the algorithm due to Seymour & Thomas (Sections 7 and 9 in [32]). (See also the results of Hicks [23] on implementations of Seymour & Thomas algorithm.) This algorithm can be implemented in $O(n^4)$ steps. Then Theorem 2 implies the proof. □

In what follows, whenever we say that a problem is *solvable for graphs of branch/treewidth bounded by some constant* we will directly assume that the input should be accompanied by the corresponding branch/tree decomposition.

Corollary 2. *Let Π be an optimization problem that is solvable on graphs of branch-width/tree-width $\leq \ell$ in time $2^{o(\ell^2)}\mathsf{poly}(n, \ell)$. Then on planar graphs problem Π is solvable in subexponential time (in $2^{o(n)}$ steps).*

In spite of its simplicity, Theorem 3 provides a general framework for obtaining subexponential algorithms for a broad range of problems. And the only thing one needs to estimate the running time of the algorithm is how fast a problem can be solved on graphs of bounded branch-width/tree-width[1]. But really surprising is that such a trivial approach provides better time estimation than many, complicated to analyze, algorithms based on separator theorems. Let us give just few examples.

Independent Set: The Independent Set problem asks, given a graph G and a non-negative integer k, whether G has an independent set, i.e. a subset S of $V(G)$ of size at most k where no edge of G has two vertices of S as an endpoint. It is well known that on graphs of tree-width ℓ Independent Set can be solved in time $O(2^{\ell}n)$ and hence on graphs of branch-width $\leq \ell$ it can be solved in time $O(2^{(3/2)\ell}n)$. Thus by Theorem 3 we obtain that Independent Set on planar graphs is solvable in $O(2^{3.182\sqrt{n}}n + n^4)$.

Dominating Set and variants: The Dominating Set problem asks, given a graph G and a non-negative integer k, whether $V(G)$ has a subset S of size at most k where any vertex in $V(G) - S$ is adjacent to some verttex in S. Dominating Set on graphs of branch-width $\leq \ell$ is solvable is time $O(2^{3\log_4 3 \cdot \ell}m)$ [13]. Thus on planar graphs, Dominating set is solvable in $O(2^{5.043\sqrt{n}}n + n^4)$.

[1] Any algorithm solving a problem on graphs of tree-width $\leq \ell$ in time $f(\ell)g(n)$ can be translated to an algorithm for graphs of branch-width $\leq \ell$ with running time $O(f(3/2\ell)g(n) + m)$ where m is the number of edges of the input graph.

Several variants of the DOMINATING SET problem can be defined if we additionally demand S to satisfy some additional property Π. Some examples of this type of problems, which are mentioned in [5], are the INDEPENDENT DOMINATING SET problem (S is an independent set), the TOTAL DOMINATING SET problem (any vertex of V is adjacent to some vertex in S), the PERFECT DOMINATING SET problem (each vertex of G is adjacent to *exactly* one vertex in S), the PERFECT INDEPENDENT DOMINATING SET problem also known as the PERFECT CODE problem, and the TOTAL PERFECT DOMINATING SET problem.

Another variant is the WEIGHTED DOMINATING SET problem in which we have a graph $G = (V, E)$ together with an integer weight function $w : V \rightarrow \mathbb{N}$ with $w(v) > 0$ for all $v \in V$. The weight of a vertex set $S \subseteq V$ is defined as $w(S) = \sum_{v \in D} w(v)$. The WEIGHTED DOMINATING SET problem asks, given a weigthed graph G and a non-negative integer k, whether G has a dominating set S where $w(S) \leq k$.

The RED-BLUE DOMINATING SET problem has as instances bipartite graphs where the bipartition is given by $V_{\mathrm{red}} \cup V_{\mathrm{blue}}$ and a non-negative integer k. The question of the problem is whether there is a subset S of V_{red} of size at most k such that every vertex of V_{blue} is adjacent to at least one vertex of S.

By [5], the INDEPENDENT DOMINATING SET, the PERFECT DOMINATING SET, the PERFECT CODE, the WEIGHTED DOMINATING SET, and the RED-BLUE DOMINATING SET problems can be solved in time $O(4^k n)$. Also, the TOTAL DOMINATING SET, and the TOTAL PERFECT DOMINATING SET, problems can be solved in time $O(5^k n)$ [5]. Thus by Theorem 3 we obtain that the INDEPENDENT DOMINATING SET, the PERFECT DOMINATING SET, the PERFECT CODE, the WEIGHTED DOMINATING SET and the RED BLUE DOMINATING SET problems can be solved in time $O(2^{6.37\sqrt{n}} n + n^4)$ for planar graphs, and that the TOTAL DOMINATING SET and the TOTAL PERFECT DOMINATING SET problems can be solved in time $O(2^{7.4\sqrt{n}} n + n^4)$ for planar graphs.

Longest Cycles and Paths: The LONGEST PATH and the LONGEST CYCLE problems asks, given a graph G and a non-negative integer k, whether G has a path (resp. cycle) of length $\geq k$. According to [8] the LONGEST CYCLE and the LONGEST PATH problems can be solved in time $O(\ell! 2^{\ell} n)$. Combining this with Theorem 3, we obtain an $O(2^{2.29\sqrt{n}(\ln n + 0.94)} n^{5/4} + n^4)$ algorithm for these problem on planar graphs[2].

[2] The calculation of the exponent in this algorithm makes use of Stirling's formula.

Min-bisection: The MIN-BISECTION problem asks, given a graph G and a non-negative integer k, whether there is a partition of V into two sets V_1 and V_2 where $||V_1| - |V_2|| \leq 1$ and such that the number of edges with endpoints in both V_1 and V_2 is at most k. From [24] , MIN-BISECTION is solvable in $O(2^{\ell}n)$ on graphs. From Theorem 3 the planar version of the problem can be solved in time $O(2^{3.182\sqrt{n}}n + n^4)$.

Kernels in digraphs: A set S of vertices in a digraph D is a *kernel* if S is independent and every vertex in $V(D) - S$ has an out-neighbor in S. The KERNEL problem asks, given a graph G and a non-negative integer k, whether G has a kernel of size $\leq k$. In [21], Gutin et al. gave a time $O(4^{\ell}n)$ algorithm for finding a kernel of size k in a digraph whose underlying[3] graph has treewidth at most ℓ This, along with Theorem 3, implies that KERNEL is solvable in time $O(2^{6.37\sqrt{n}}n + n^4)$.

H**-coloring:** Let H be a graph with h vertices. The H-COLORING problem asks, given a graph G, whether there exists a *homomorphism* from G to H, i.e. a mapping $\sigma : V(G) \to V(H)$ such that for any edge $\{v, u\} \in E(G)$, $\{\sigma(v), \sigma(u)\}$ is also an edge of H. In [15] is given an algorithm that, given a tree decomposition of G of width at most ℓ, solves the H-COLORING problem in time $O(h^{\ell+1}\ell n)$. From Theorem 3, the planar version of the H-COLORING problem can be solved in time $O(2^{\log h \cdot 2.12\sqrt{n}}hn^{3/2} + n^4)$.

H**-cover:** Let H be a graph with h vertices. The H-COLORING problem asks, given a graph G, whether there exists a *homomorphism* σ from G to H that, when restricted to the closed neighbourhood $N_G[v]$ of an arbitrary vertex v of G, is an isomorphism to $N_H[\sigma(v)]$. By [33], the H-COVER problem is solvable in time $O(n2^{3\ell h})$. From Theorem 3, the planar version of the H-COVER problem can be solved in time $O(2^{9.546\sqrt{n}h}n + n^4)$.

r**-center:** The r-CENTER problem is a natural generalization of *Dominating Set*. We define the *r-neighborhood* of a set $S \subseteq V(G)$, denoted by $N_G^r(S)$, to be the set of vertices of G at distance at most r from at least one vertex of S. The r-CENTER problem asks, given a graph G and a non-negative integer k, whether there exists a set S of vertices (we call them *centers*) of size at most k such that $N_G^r(S) = V(G)$. the r-CENTER problem is solvable in time $O((2r+1)^{\frac{3}{2}\cdot\ell}m)$ on graphs of branch-width $\leq \ell$ [13]. Combining this with Theorem 3, we obtain an $O((2r+1)^{3.182\sqrt{n}}n + n^4)$ algorithm for the planar version of the problem.

[3] The underlying graph of a digraph is the graph obtained if we replace the directed edges by simple edges.

More generally, it seems that almost every natural problem expressible in MSOL is solvable in time $O(c^{\ell}n^{O(1)})$, $O(\ell^{\ell}n^{O(1)})$ or $O(\ell!c^{\ell}n^{O(1)})$, and by Corollary 2 is solvable in subexponential time on planar graphs. Examples of such problems solvable in $O(c^{\ell}n^{O(1)})$, where c is a small constant are VERTEX FEEDBACK SET, DISJOINT CYCLES, FACE COVER. EDGE DOMINATING SET, CLIQUE TRANSVERSAL, and MAXIMAL MATCHING (see [10, 14] for definitions and algorithms). For all these problems, Corollary 2 provides subexponential algorithms with small hidden constants. We do not proceed to a detailed analysis of each individual case as we believe that the examples given so far are enough to give the main flavour of our approach.

Actually, one can further strengthen the conditions of Corollary 2 towards extending the framework where subexponential algorithms are possible. Indeed, it is enough to have a time $(\mathsf{poly}(\ell, n))^{o(\ell^2)}$ algorithm for the problem Π for graphs of treewidth/branchwidth at most ℓ. Notice that such problems are not necessarily expressible in MSOL. We will proceed with an example related to problems of finding optimum non-preemptive multicolorings.

Non-preemptive multicoloring: Let G be a graph and p some positive integer. A *non-preemptive p-multicoloring*[4] of G is an assignment ψ mapping each vertex of v to some set of consecutive positive integers, each not bigger than p, such that adjacent vertices receive non-intersecting sets. The *sum* of a multicoloring ψ is equal to $\Sigma_{v\in V} \max_{i\in\psi(v)} i$. The *makespan* of a multicoloring ψ is equal to $\max_{v\in V} \max_{i\in\psi(v)} i$. The MINIMUM SPAN NON-PREEMPTIVE p-MULTICOLORING problem asks, given a graph G and a non-negative integer k, whether G has a non-preemptive p-multicoloring with span at most k. The MINIMUM MAKESPAN NON-PREEMPTIVE p-MULTICOLORING problem is defined analogously by asking for a non-preemptive multicoloring with makespan at most k.

According to [22], MINIMUM SPAN NON-PREEMPTIVE p-MULTICOLORING and MINIMUM MAKESPAN NON-PREEMPTIVE p-MULTICOLORING can be solved in time $O(n \cdot (\ell p \log n)^{\ell+1})$ for graphs with tree-width $\leq \ell$. Therefore, both of them can be solved in time $O(pn^{3/2} \log n \cdot 2^{1.15\cdot\log p \log n \log\log n\sqrt{n}} + n^4)$ on planar graphs.

[4] The multicoloring problem has numerous aplications in job scheduling on multiprocessor systems [22], traffic intersection control [9], compiler design and VLSI routing [31].

4.2 Algorithms for parameterized problems

Similar ideas work for parameterized problems. Let $\mathcal{L}$ be a parameterized problem, i.e. $\mathcal{L}$ consists of pairs (I,k) where k is the *parameter* of the problem. *Reduction to linear problem kernel* is the replacement of problem inputs (I,k) by a reduced problem with inputs (I',k') (linear kernel) with constants c_1, c_2 such that $k' \le c_1 k$, $|I'| \le c_2 k'$ and $(I,k) \in \mathcal{L} \Leftrightarrow (I',k') \in \mathcal{L}$. (We refer to Downey & Fellows [16] for discussions on fixed parameter tractability and the ways of constructing kernels.)

Theorem 4. *Let $\mathcal{L}$ be a parameterized problem (I,k) (here I can be a graph, hypergraph or matroid) such that*
— There is a linear problem kernel computable in time $T_{kernel}(|I|,k)$ with constants c_1, c_2 and such that an optimal branch decomposition of the kernel is computable in time $T_{bw}(|I'|)$.
— On graphs (hypergraphs, matroids) of branch-width $\le \ell$ and ground set of size n the problem $\mathcal{L}$ can be solved in $O(2^{c_3 \ell} n)$, where c_3 is a constant.
— $\mathbf{bw}(I') \le c_4\sqrt{k}$, where c_4 is a constant. Then $\mathcal{L}$ can be solved in time $O(2^{c_3 c_4 \sqrt{k}} k + T_{bw}(|I'|) + T_{kernel}(|I|,k))$.

Proof. The algorithm works as follows. First we compute a linear kernel in time $T_{kernel}(|I|,k)$. Then we construct a branch decomposition of the kernel in $T_{bw}(|I'|)$ steps. The size of the kernel is at most $c_1 c_2 k = O(k)$. The branch-width of the kernel is at most $c_4\sqrt{k}$ and it takes $O(2^{c_3 c_4 \sqrt{k}} k + T_{bw}(|I'|) + T_{kernel}(|I|,k))$ to solve the problem. □

Let us give some examples, where Theorem 4 provides *proven* better bounds for different parameterized problems.

The Planar k-Vertex Cover problem is the task to compute, given a planar graph G and a positive integer k, a vertex cover of size k or to report that no such a set exists. A linear problem kernel of size $2k$ (with constants $c_1 = 1$ and $c_2 = 2$) for the k-Vertex Cover problem (not necessary planar) was obtained by Chen et al. [11]. The running time of the algorithm constructing a kernel of a graph on n vertices is $O(kn + k^3)$. So in this case $T_{kernel}(|I|,k) = O(kn+k^3)$. It is well known that the Vertex Cover problem on graphs on n vertices and with bounded tree-width $\le \ell$ can be solved in $O(2^{\ell} n)$ time. The dynamic programming algorithm for the Vertex Cover on graphs with bounded tree-width can be easy translated to the dynamic programming algorithm for graphs with bounded branch-width with running time $O(2^{3/2\ell} m)$, where m is the number of edges in a graph, and we omit it here. For planar graphs $2^{3/2\ell}m = O(2^{3/2\ell}n)$, thus $c_3 \leq 3/2$.

From the constructions used in the reduction algorithm of Chen et al. [11] it follows that if G is a planar graph then the kernel graph is also planar. To compute an optimal branch decomposition of a planar graph one can use the algorithm due to Seymour & Thomas [32]. This algorithm (applied to the kernel graph) can be implemented in $O(k^4)$ steps. The kernel graph I' has at most $2k$ vertices. Then by Theorem 2, $c_4 \leq \sqrt{4.5}\sqrt{2} = 3$. Thus by making use of Theorem 4, we conclude that Planar k-Vertex Cover can be solved in $O(k^4 + 2^{4.5\sqrt{k}}k + kn)$.

A *k-dominating* set D of a graph G is a set of k vertices such that every vertex outside D is adjacent to a vertex of D. The Planar k-Dominating Set problem is the task to compute, given a planar graph G and a positive integer k, a k-dominating set or to report that no such a set exists.

Alber, Fellows & Niedermeier [2] show that the Planar Dominating Set problem admits a linear problem kernel. (The size of the kernel is $335k$.) This reduction can be performed in $O(n^3)$ time. Dominating Set problem on graphs of branch-width $\leq \ell$ can be solved in $O(2^{3\log_4 3 \cdot \ell}m)$ steps [20]. Thus $c_3 \leq 3\log_4 3$. It is proved in [20] that for every planar graph G with dominating set k, the branch-width of G is at most $3\sqrt{4.5}\sqrt{k}$, i.e. $c_4 \leq 3\sqrt{4.5}$. Then by Theorem 4, Planar Dominating set can be solved in $O(2^{15.13\sqrt{k}}k + n^3 + k^4)$.

4.3 Other problems and generalizations.

Our ideas can be adapted to different problems by using the bounds and tree-width (branch-width) based algorithms in the same fashion as it is done in [1, 3, 10, 14]. That way, our upper bound implies the construction of faster algorithms for a series of problems when their inputs are restricted to planar graphs. As a sample we mention parameterized versions of the following problems: Independent Dominating Set, Perfect Dominating Set, Perfect Code, Weighted Dominating Set, Total Dominating Set, Edge Dominating Set, Face Cover, Vertex Feedback Set, Minimum Maximal Matching, Clique Transversal Set, Disjoint Cycles, and Digraph Kernel (see [1, 3, 10, 14] for the exact definitions).

Finally let us note that our upper bound for treewidth holds not only on planar graphs but on different generalizations of planar graphs. This follows directly from the results of [14] and implies an exponential speed-up of all the aforementioned problems on certain classes of non-planar graphs such as $K_{3,3}$-minor-free or K_5-minor-free graphs.

References

1. J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.
2. J. Alber, M. R. Fellows, and R. Niedermeier. Efficient data reduction for dominating set: A linear problem kernel for the planar case. In *8th Scandinavian Workshop on Algorithm Theory—SWAT 2002 (Turku, Finland)*, pages 150–159. Springer, vol. 2368, Berlin, 2002.
3. J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: Exponential speed-up for planar graph problems. In *Electronic Colloquium on Computational Complexity (ECCC)*. Germany, 2001.
4. J. Alber, H. Fernau, and R. Niedermeier. Graph separators: a parameterized view. *Journal of Computer and System Sciences (to appear)*, 2003.
5. J. Alber and R. Niedermeier. Improved tree decomposition based algorithms for domination-like problems. In *Latin American Theoretical Informatics—LATIN 2002 (Cancun, Mexico)*, volume 2286, pages 613–627. Springer, Lecture Notes in Computer Science, Berlin, 2002.
6. N. Alon, P. Seymour, and R. Thomas. A separator theorem for nonplanar graphs. *J. Amer. Math. Soc.*, 3(4):801–808, 1990.
7. N. Alon, P. Seymour, and R. Thomas. Planar separators. *SIAM J. Discrete Math.*, 7(2):184–193, 1994.
8. H. L. Bodlaender. On linear time minor tests with depth-first search. *J. Algorithms*, 14(1):1–23, 1993.
9. D. Bullock and C. Hendrickson. Roadway traffic control software. *IEEE Transactions on Control Systems Technology*, 2:255–264, 1994.
10. M.-S. Chang, T. Kloks, and C.-M. Lee. Maximum clique transversals. In *27th International Workshop on Graph-Theoretic Concepts in Computer Science—WG 2001 (Boltenhagen near Rostock, Germany)*, pages 300–310, 2001.
11. J. Chen, I. A. Kanj, and W. Jia. Vertex cover: further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.
12. E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, P. Raghavan, and U. Schöning. A deterministic $(2-2/(k+1))^n$ algorithm for k-SAT based on local search. *Theoretical Computer Science*, 289:69–83, 2002.
13. E. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Fixed-parameter algorithms for the (k,r)-center in planar graphs and map graphs. In *13th International Colloquium on Automata, Languages and Programming—ICALP 2003 (Eindhoven, The Netherlands)*, volume 2719, pages 829–844. Springer, Lecture Notes in Computer Science, 2003.
14. E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos. Exponential speedup of fixed parameter algorithms on $K_{3,3}$-minor-free or K_5-minor-free graphs. In *13th Annual International Symposium on Algorithms and Computation—ISAAC 2002 (Vancouver, Canada)*, volume 2518, pages 262–273. Springer, Lecture Notes in Computer Science, Berlin, 2002.
15. J. Diaz, M. Serna, and D. M. Thilikos. Counting H-colorings of partial k-trees. *Theoretical Computer Science*, 281:291–309, 2002.
16. R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
17. D. Eppstein. Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms—SODA 2001 (Washington, DC)*, pages 329–337, Philadelphia, PA, 2001. SIAM.

18. U. Feige. Coping with the NP-hardness of the graph bandwidth problem. In *The 6th Scandinavian Workshop on Algorithm Theory—SWAT 2000 (Bergen)*, volume 1851, pages 10–19, Berlin, 2000. Springer, Lecture Notes in Computer Science.
19. F. V. Fomin and D. M. Thilikos. New upper bounds on the decomposability of planar graphs and fixed parameter algorithms. Technical report, Universitat Politècnica de Catalunya, Spain, 2002.
20. F. V. Fomin and D. M. Thilikos. Dominating sets in planar graphs: Branch-width and exponential speed-up. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms—SODA 2003 (Baltimore, MD)*, pages 168–177, 2003.
21. G. Gutin, T. Kloks, and C. M. Lee. Kernels in planar digraphs. In *Optimization Online*. Mathematical Programming Society, Philadelphia, 2001.
22. M. M. Halldórsson and G. Kortsarz. Tools for multicoloring with applications to planar graphs and partial k-trees. *Journal of Algorithms*, 42(2):334–366, 2002.
23. I. V. Hicks. *Branch Decompositions and their applications*. PhD thesis, Rice University, 2000.
24. K. Jansen, M. Karpinski, A. Lingas, and E. Seidel. Polynomial time approximation schemes for Max-Bisection on planar and geometric graphs. In *8th Annual Symposium on Theoretical Aspects of Computer Science—STACS 2001 (Dresden)*, volume 2010, pages 365–375. Springer, Lecture Notes in Computer Science, Berlin, 2001.
25. I. Kanj and L. Perković. Improved parameterized algorithms for planar dominating set. In *Mathematical Foundations of Computer Science—MFCS 2002 (Warsaw, Poland)*, volume 2420, pages 399–410. Springer, Lecture Notes in Computer Science, Berlin, 2002.
26. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36:177–189, 1979.
27. R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9:615–627, 1980.
28. N. Robertson and P. D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.
29. N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.
30. N. Robertson and P. D. Seymour. Graph minors. XI. Circuits on a surface. *J. Combin. Theory Ser. B*, 60(1):72–106, 1994.
31. X. S. S. Nicoloso, Majid Sarrafzadeh. On the sum coloring problem on interval graphs. *Algorithmica*, 23(2):109–126, 1999.
32. P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.
33. J. A. Telle and A. Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.*, 10(4):529–550, 1997.
34. G. J. Woeginger. Exact algorithms for NP-hard problems: A survey. In *5th Int. Workshop Combinatorial Optimization - Eureka*, volume 2570, pages 185–207. Springer, Lecture Notes in Computer Science, 2003.