REPORTS
IN
INFORMATICS

ISSN 0333-3590

Sphere Cut Branch Decomposition: Subexponential Algorithms on Planar Graphs

Frederic Dorn and Fedor V. Fomin

REPORT NO 296 **April 2005**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2005-296.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Sphere Cut Branch Decomposition: Subexponential Algorithms on Planar Graphs

Frederic Dorn and Fedor V. Fomin*

Department of Informatics, University of Bergen, N-5020 Bergen, Norway,
{frederic.dorn, fedor.fomin}@ii.uib.no

Abstract Divide-and-conquer strategy based on variations of Lipton-Tarjan planar separator theorem is one of the most common approaches for solving planar graph problems for more than 20 years. We present a new framework for designing fast subexponential exact and parameterized algorithms on planar graphs. Our approach is based on geometric properties of planar branch decompositions obtained by Seymour & Thomas combined with new techniques of dynamic programming on planar graphs. To compare with divide-and-conquer algorithms, the main advantages of our method are a) it is a generic method which allows to attack broad classes of problems; b) the obtained algorithms provide a better worst case analysis. To exemplify our approach we show how to obtain an $O(2^{10.8224\sqrt{n}}n^{3/2} + n^4)$ time algorithm solving weighted Planar Graph TSP. We observe how our technique can be used to solve Hamiltonian Cycle on planar graphs with n vertices in time $O(2^{6.982\sqrt{n}}n^{3/2} + n^4)$. Our approach can be used to design parameterized algorithms as well. For example we introduce the first $2^{O(\sqrt{k})}k^{O(1)} \cdot n^{O(1)}$ time algorithm for parameterized Planar k−cycle by showing that for a given k we can decide if a planar graph on n vertices has a cycle of length $\geq k$ in time $O(2^{13.6\sqrt{k}}\sqrt{k}\,n + n^4)$.

Keywords: Exact and parameterized algorithms, planar graphs, treewidth, branchwidth, traveling salesman problem, Hamiltonian cycle.

1 Introduction

The celebrated Lipton-Tarjan planar separator theorem [16] is one of the basic approaches to obtain algorithms with subexponential running time for many problems on planar graphs [17]. The usual running time of such algorithms is $2^{O(\sqrt{n})}$ or $2^{O(\sqrt{n}\log n)}$ however the constants hidden in big-Oh of the exponent are the serious obstacle for practical implementations. During the last few years a lot of work has been done to improve the running time of divide-and-conquer type algorithms [3,4].

One of the possible alternatives to divide-and-conquer algorithms on planar graphs was suggested by Fomin & Thilikos [12]. The idea of this approach is very simple: compute treewidth (or branchwidth) of a planar graph and then use the well developed machinery of dynamic programming on graphs of bounded treewidth (or branchwidth)[5]. For example, given a branch decomposition of width ℓ of a graph G on n vertices, it can be shown that the maximum independent set of G can be found in time $O(2^{\frac{3\ell}{2}}n)$. The branchwidth of a planar graph G is at most $2.122\sqrt{n}$ and it can be found in time $O(n^4)$ [20]. Putting all together, we obtain $O(2^{3.182\sqrt{n}}n + n^4)$ time algorithm solving independent set problem on planar graphs. Note that planarity come into play twice in this approach: First in the upper bound on the branchwidth of a graph and second in the polynomial time algorithm constructing an optimal branch decomposition. A similar approach combined with the results

* Supported by Norges forskningsråd project 160778/V30.

from Graph Minors [18] works for many parameterized problems on planar graphs [8]. However, if we try to use such an approach to solve, for example, Hamiltonian cycle in time $2^{O(\sqrt{n})}n^{O(1)}$ we get stuck because of the following obstacle: All known algorithms solving the Hamiltonian cycle problem on graphs of tree-width ℓ require $2^{O(\ell \log \ell)}n^{O(1)}$ steps and we result with a $2^{O(\sqrt{n}\log n)}n^{O(1)}$ algorithm on planar graphs. In this paper we show how to get rid of the logarithmic factor in the exponent for a number of problems. The main idea to speed-up algorithms obtained by the branch decomposition approach is to exploit planarity for the third time. In addition to upper bound and polynomial time algorithm for branchwidth, for the first planarity is used in dynamic programming on graphs of bounded treewidth. Our results are based on very deep results of Seymour & Thomas [20] on geometric properties of planar branch decompositions. Loosely speaking, the results of Seymour & Thomas imply that for a graph G embedded on a sphere Σ, some branch decompositions can be seen as decompositions of Σ into discs (or sphere cuts). We are the first describing such geometric properties of so-called sphere cut branch decompositions. Sphere cut branch decompositions seem to be an appropriate tool for solving a variety of planar graph problems. To demonstrate the power of the new method we apply it to the following problems.

Planar Graph TSP. The Traveling Salesman Problem (TSP) is one of the most attractive problems in Computer Science and Operations Research. For several decades, almost every new algorithmic paradigm was tried on TSP including approximation algorithms, linear programming, local search, polyhedral combinatorics, and probabilistic algorithms [15]. One of the first known exact exponential time algorithms is actually the algorithm of Held and Harp [13] solving TSP on n cites in time $2^n n^{O(1)}$ by making use of dynamic programming. For some special cases like Euclidean TSP (when the n cites are points in the Euclidean plane and the distances between the cites are Euclidean distances), by exploiting planar separator structures several researchers independently obtained subexponential algorithms of running time $2^{O(\sqrt{n}\cdot logn)}n^{O(1)}$ (see e.g. [14]). Smith & Wormald [21] succeed to generalize these results to d-space and the running time of their algorithm is $2^{d^{O(d)}} \cdot 2^{O(dn^{1-1/d}\log n)} + 2^{O(d)}$. Until very recent there was no known $2^{O(\sqrt{n})}n^{O(1)}$-time algorithm even for a very special case of TSP, namely Planar Hamiltonian Cycle. Recently, Deĭneko et al. [7] obtained a divide-and-conquer type algorithm of running time $2^{O(\sqrt{n})}n^{O(1)}$ for Planar Hamiltonian Cycle. In this paper we use sphere cut branch decompositions to obtain the first $c^{O(\sqrt{n})}n^{O(1)}$ time algorithm for another variant of TSP, Planar Graph TSP, which for a given weighted planar graph G is a TSP with distance metric the shortest path metric of G. Since Planar Hamiltonian Cycle is a special case of Planar Graph TSP it also can be solved in $2^{O(\sqrt{n})}n^{O(1)}$ by our method. Though our ideas to solve Planar Graph TSP can be applied on the divide-and-conquer approach from [7], dynamic programming provides much better and delicate worst case running time analysis. And the huge difference is hidden in the big-O of the exponent. For example, our branch decomposition algorithm solves Planar Hamiltonian Cycle in time $O(2^{6.982\sqrt{n}}n^{3/2} + n^4)$, while the divide-and-conquer algorithm of Deĭneko et al. [7] for graphs with $n \leq 1152$ vertices tries brute force and for graphs on $n > 1152$ has running time roughly $2^{126\sqrt{n}}n^{O(1)}$. Of course Deĭneko et al. put no efforts in optimizing the running of their algorithm (their goal was only to get rid of logarithmic factor in the exponent), but even with more careful analyses of the divide-and-conquer algorithm it is difficult to obtain small constants in the exponent because of the recursive nature of such an algorithm.

Parameterized Planar k-cycle. The last ten years were the evidence of a rapid development of a new branch of computational complexity: Parameterized Complexity. (See the book of Downey & Fellows [9].) Roughly speaking, a parameterized problem with parameter k is *fixed parameter tractable* if it admits an 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 is an exponential function, e.g. $f(k) = 2^{O(k)}$. However, it appears, that for a large variety of planar graph problems algorithms with growth of the form $f(k) = 2^{O(\sqrt{k})}$ are possible. During the last two years much attention was paid to the construction of algorithms with running time $2^{O(\sqrt{k})}$ 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 $O(2^{70\sqrt{k}}n)$ for the PLANAR DOMINATING SET problem. Different fixed parameter algorithms for solving problems on planar and related graphs are discussed in [3,4,8]. In the PLANAR k-CYCLE problem a parameter k is given and the question is if there exists a cycle of length at least k in a planar graph. There are several ways to obtain algorithms solving different generalizations of PLANAR k-CYCLE in time $2^{O(\sqrt{k}\log k)}n^{O(1)}$, one of the most general results is Eppstein's algorithm [10] solving the PLANAR SUBGRAPH ISOMORPHISM problem with pattern of size k in time $2^{O(\sqrt{k}\log k)}n$. By making use of sphere cut branch decompositions we succeed to find $O(2^{13.6\sqrt{k}}k\,n + n^4)$ time algorithm solving PLANAR k-CYCLE.

2 Geometric Branch Decompositions of Σ-plane Graphs

In this section we introduce our main technical tool, sphere cut branch decomposition but first we give some definitions.

Let Σ be a sphere $(x, y, z\colon x^2 + y^2 + z^2 = 1)$. By a *$\Sigma$-plane* graph G we mean a planar graph G with the vertex set $V(G)$ and the edge set $E(G)$ drawn (without crossing) in Σ. Throughout the paper, we denote by n the number of vertices of G. To simplify notations, we usually do not distinguish between a vertex of the graph and the point of Σ used in the drawing to represent the vertex or between an edge and the open line segment representing it. An O-arc is a subset of Σ homeomorphic to a circle. An O-arc in Σ is called *noose* of a Σ-plane graph G if it meets only vertices of G. The length of a noose O is $|O \cap V(G)|$, i.e. is the number of vertices it meets. Every noose O bounds two open discs Δ_1, Δ_2 in Σ, i.e. $\Delta_1 \cap \Delta_2 = \emptyset$ and $\Delta_1 \cup \Delta_2 \cup O = \Sigma$.

Branch Decompositions and Carving Decompositions. A *branch decomposition* $\langle T, \mu\rangle$ of a graph G consists of an un-rooted ternary (i.e. with all vertices of degree three or one) tree T and a bijection $\mu : L \to E(G)$ from the set L of all leaves of T to the edge set of G. We define for every edge e of T the *middle set* $\mathrm{mid}(e) \subseteq V(G)$ as follows: Let T_1 and T_2 be the two connected components of $T \setminus \{e\}$. Then let G_i be the graph induced by the edge set $\{\mu(f) : f \in L \cap V(T_i)\}$ for $i \in \{1, 2\}$. The *middle set* is the intersection of the vertex sets of G_1 and G_2, i.e., $\mathrm{mid}(e) := V(G_1) \cap V(G_2)$. The *width* bw of $\langle T, \mu\rangle$ is the maximum order of the middle sets over all edges of T, i.e., $\mathrm{bw}(\langle T, \mu\rangle) := \max\{|\,\mathrm{mid}(e)|\colon e \in T\}$. An optimal branch decomposition of G is defined by the tree T and the bijection μ which together provide the minimum width, the *branchwidth* $\mathrm{bw}(G)$.

A *carving decomposition* $\langle T, \mu\rangle$ is similar to a branch decomposition, only with the difference that μ is the bijection between the leaves of the tree and the *vertex set* of the graph. For an edge e of T, the counterpart of the middle set, called *cut set* $\mathrm{cut}(e)$, contains the edges of the graph with end vertices in the leaves of both subtrees. The counterpart of branchwidth is carvingwidth.

We will need the following result.

Proposition 1 ([12]). *For any planar graph G, $\mathrm{bw}(G) \leq \sqrt{4.5n} \leq 2.122\sqrt{n}$.*

Sphere Cut Branch Decompositions. For a Σ-plane graph G, we define a *sphere cut branch decomposition* or *sc-branch decomposition* $\langle T, \mu, \pi\rangle$ as a branch decomposition such that for every edge e of T and the two subgraphs G_1 and G_2 induced by the leaves of the subtrees of $T \setminus \{e\}$, there exists a noose O_e bounding the two open discs Δ_1 and Δ_2 such that $G_i \subseteq \Delta_i \cup O_e$, $1 \leq i \leq 2$. Thus O_e meets G only in $\mathrm{mid}(e)$ and its length is $|\mathrm{mid}(e)|$. Clockwise traversing of O_e in the drawing G defines the cyclic ordering π of $\mathrm{mid}(e)$. We always assume that in an sc-branch decomposition the vertices of every middle set $\mathrm{mid}(e) = V(G_1) \cap V(G_2)$ are enumerated according to π.

The following theorem provides us with the main technical tool. It follows almost directly from the results of Seymour & Thomas [20]. Since this is not explicitly mentioned in [20], we provide some explanations below.

Theorem 1. *Let G be a connected Σ-plane graph of branchwidth $\leq \ell$ without vertices of degree one. There exists an sc-branch decomposition of G of width $\leq \ell$ and such a branch decomposition can be constructed in time $O(n^4)$.*

Proof. Let G be a Σ-plane graph of branchwidth $\leq \ell$ and with minimal vertex degree ≥ 2.

The *medial graph* M_G of G has vertex set $E(G)$, and, for every vertex $v \in V(G)$ there is a cycle C_v in M_G with the following properties:

1. The cycles C_v $(v \in V(G))$ of M_G are mutually edge-disjoint and have union M_G;
2. For each $v \in V(G)$, let the neighbors of v in G be $w_1, \ldots, w_t$ enumerated according to the cyclic order of the edges $\{v, w_1\}, \ldots, \{v, w_t\}$ in the drawing of G; then C_v has vertex set $\{w_1, \ldots, w_t\}$, and in C_v w_{i-1} is adjacent to w_i $(1 \leq i \leq t)$, where w_0 means w_t.

In a *bond* carving decomposition of a graph, every cut set is a bond of the graph, i.e., every cut set is a minimal cut. Seymour and Thomas ((5.1) and (7.2) [20]) show that a Σ-plane graph G without vertices of degree one is of branchwidth $\leq \ell$ if and only if M_G has a bond carving decomposition of width $\leq 2\ell$. They also show (Algorithm (9.1) in [20]) how to construct an optimal bond carving decompositions of the medial graph M_G in time $O(n^4)$. A bond carving decomposition $\langle T, \mu \rangle$ of M_G is also branch decomposition of G (vertices of M_G are the edges of G) and it can be shown (see the proof of (7.2) in [20]) that for every edge e of T if the cut set $\mathrm{cut}(e)$ in M_G is of size $\leq 2\ell$, then the middle set $\mathrm{mid}(e)$ in G is of size $\leq \ell$. It is well known that the edge set of a minimal cut forms a cycle in the dual graph. The dual graph of a medial graph M_G is the *radial graph* R_G. In other words, R_G is a bipartite graph with the bipartition $F(G) \cup V(G)$. A vertex $v \in V(G)$ is adjacent in R_G to a vertex $f \in F(G)$ if and only if the vertex v is incident to the face f in the drawing of G. Therefore, a cycle in R_G forms a noose in G.

To summarize, for every edge e of T, $\mathrm{cut}(e)$ is a minimal cut in M_G, thus $\mathrm{cut}(e)$ forms a cycle in R_G (and a noose O_e in G). Every vertex of M_G is in one of the open discs Δ_1 and Δ_2 bounded by O_e. Since O_e meets G only in vertices, we have that $O_e \cap V(G) = \mathrm{mid}(e)$. Thus for every edge e of T and the two subgraphs G_1 and G_2 of G formed by the leaves of the subtrees of $T \setminus \{e\}$, O_e bounds the two open discs Δ_1 and Δ_2 such that $G_i \subseteq \Delta_i \cup O_e$, $1 \leq i \leq 2$.

Finally, with a given bond carving decomposition $\langle T, \mu \rangle$ of the medial graph M_G, it is straightforward to construct cycles in R_G corresponding to $\mathrm{cut}(e)$, $e \in E(T)$, and afterwards to compute ordering π of $\mathrm{mid}(e)$ in linear time. ☐

3 Planar Graph TSP

In the Planar Graph TSP we are given a weighted Σ-plane graph $G = (V, E)$ with weight function $w \colon E(G) \to N$ and we are looking for a shortest closed walk that visits all vertices of G at least once. Equivalently, this is TSP with distance metric the shortest path metric of G. It is easy to show that a shortest closed walk passes through each edge at most twice. Thus every shortest closed walk in G corresponds to the minimum Eulerian subgraph in the graph G' obtained from G by adding to each edge a parallel edge. Every vertex of an Eulerian graph is of even degree, which brings us to another equivalent formulation of the problem. A labeling $\mathcal{E} : E(G) \rightarrow \{0, 1, 2\}$ is *Eulerian* if the subgraph $G_{\mathcal{E}}$ of G formed by the edges with positive labels is a connected spanning subgraph and for every vertex $v \in V$ the sum of labels assigned to edges incident to v is even. Thus the problem of optimum salesman tour is equivalent to finding an Eulerian labeling $\mathcal{E}$ minimizing $\sum_{e \in E(G)} \mathcal{E}(e) \cdot w(e)$.

Instead of dealing with salesman tours directly, it is preferable for us to work with Eulerian labelings. For an edge labeling $\mathcal{E}$ and vertex $v \in V(G)$ we define the *$\mathcal{E}$-degree* $deg_{\mathcal{E}}(v)$ of v as the sum of labels assigned to the edges incident to v.

Let G be a Σ-plane graph and let $\langle T, \mu, \pi \rangle$ be a sc-branch decomposition of G of width ℓ. We root T by choosing arbitrarily an edge e, subdivide e by inserting a new node s. Let e', e'' be the new edges then we set $\mathrm{mid}(e') = \mathrm{mid}(e)$ and $\mathrm{mid}(e'') = \mathrm{mid}(e)$. Create a new node *root* r and connect it to s and set $\mathrm{mid}(\{r, s\}) = \emptyset$. Each node v of T now has one adjacent edge on the path from v to r, called *parent edge* e_P, and two adjacent edges towards the leaves, called *left child* e_L and *right child* e_R. For every edge e of T, we call the subtree towards the leaves the *lower part* and the rest the *residual part* concerning to e. We call the subgraph G_e induced by the leaves of the lower part of e the *subgraph rooted at* e.

Let e be an edge of T and let O_e be the corresponding noose in Σ. Note that noose O_e partitions Σ into two discs, one of which, Δ_e, contains G_e. We call a labeling $\mathcal{P}[e]\colon E(G_e) \rightarrow \{0, 1, 2\}$ a *partial Eulerian labeling* if the subgraph $G_{\mathcal{P}[e]}$ induced by the edges with positive labels satisfies the following properties

- Every connected component of $G_{\mathcal{P}[e]}$ has a vertex in O_e for $e \neq \{r, s\}$; For $e = \{r, s\}$, $G_{\mathcal{P}[e]}$ is connected.
- For every vertex $v \in V(G_e) \setminus O_e$, the $\mathcal{P}[e]$-degree $deg_{\mathcal{P}[e]}(v)$ of v (i.e. the sum of labels assigned by $\mathcal{P}[e]$ to the edges incident to v) is even and positive.

The weight of a partial Eulerian labeling $\mathcal{P}$ is $\sum_{f \in E(G_e)} \mathcal{P}(f) \cdot w(f)$. Note that every partial Eulerian labeling of $G_{\{r,s\}}$ is also Eulerian labeling.

To perform dynamic programming we need to keep for every edge e of T the information on which vertices of the connected components of $G_{\mathcal{P}[e]}$ of all possible partial Eulerian labelings $\mathcal{P}[e]$ hit $O_e \cap V(G)$ and for every vertex $v \in O_e \cap V(G)$ the information if $deg_{\mathcal{P}[e]}(v)$ is either 0, or odd, or even and positive.

And here the geometric properties of sc-branch decompositions come into play. For a partial Eulerian labeling $\mathcal{P}[e]$ let C be a component of $G_{\mathcal{P}[e]}$ with at least two vertices in noose O_e. We scan the vertices of $V(C) \cap O_e$ according to the ordering π and mark with index '[' the first and with ']' the last vertex of C on O_e. We also mark by '□' the other 'inner' vertices of $V(C) \cap O_e$. If C has only one vertex in O_e, we mark this vertex by '0'. Now the trick is that if we mark in such a way all vertices of $V(G_{\mathcal{P}[e]}) \cap O_e$, then given the obtained sequence with marks '[', ']', '□', and '0', one can decode the complete information on which vertices of each connected component of $V(G_{\mathcal{P}[e]})$ hit O_e. This is possible because O_e is bounding the disc Δ_e and the graph $G_{\mathcal{P}[e]}$ is in Δ_e. Since none of the connected components of $G_{\mathcal{P}[e]}$ cross, we have that for any two components C_1 and C_2 if the opening bracket '[' of C_1 goes before the opening bracket of C_2 in π, then the closing bracket ']' of C_2 is before the closing bracket of C_1. Thus every such state must be an *algebraic term* with the indices '[' being the opening and ']' the closing bracket (with '□' and '0' representing a possible term inside the brackets).

For an edge e of T and corresponding noose O_e, the state of dynamic programming is specified by an ordered ℓ-tuple $\boldsymbol{t}_e := (v_1, \ldots, v_\ell)$. Here, the variables $v_1, \ldots, v_\ell$ correspond to the vertices of $O_e \cap V(G)$ taken according to the cyclic order π with an arbitrary first vertex. (To simplify notations, we do not distinguish the names variables and the corresponding vertices.) This order is necessary for a well-defined encoding for the states when allowing $v_1, \ldots, v_\ell$ to have one of the seven values: 0, $1_{[}$, $1_{\square}$, $1_{]}$, $2_{[}$, $2_{\square}$, $2_{]}$. Hence, there are at most $O(7^\ell |V(G)|)$ states. For every state, we compute a value $W_e(v_1, \ldots, v_\ell)$. This value is equal to W if and only if W is the minimum weight of a partial Eulerian labeling $\mathcal{P}[e]$ such that:

1. For every connected component C of $G_{\mathcal{P}[e]}$ with $|C \cap O_e| \geq 2$ the first vertex of $C \cap O_e$ in π is represented by $1_{[}$ or $2_{[}$ and the last vertex is represented by $1_{]}$ or $2_{]}$. All other vertices of $C \cap O_e$ are represented by $1_{\square}$ or $2_{\square}$. For every vertex v marked by by $1_{[}$, $1_{\square}$, or $1_{]}$ the parity of $deg_{\mathcal{P}[e]}(v)$ is 1 and for every vertex v marked by by $2_{[}$, $2_{\square}$, or $2_{]}$ $deg_{\mathcal{P}[e]}(v)$ is positive and even.
2. For every connected component C of $G_{\mathcal{P}[e]}$ with $v = C \cap O_e$, v is represented by 0. (Note that since for every $w \in V(G_e) \setminus O_e$ it holds that $deg_{\mathcal{P}[e]}(w)$ is even so must $deg_{\mathcal{P}[e]}(v)$.)
3. Every vertex $v \in (V(G_e) \cap O_e) \setminus G_{\mathcal{P}[e]}$ is marked by 0. (Note that the vertices of the last two items can be treated in the same way in the dynamic programming.)

We put $W = +\infty$ if no such labeling exists. For an illustration of a partial Eulerian labeling see the upper part of Figure 6 in Appendix.

To compute an optimal Eulerian labeling we perform dynamic programming over middle sets. Note that we often do not distinguish between mid(e) and $O_e \cap V(G)$. We start at the leaves of T and work 'bottom-up' processing the subgraphs rooted at the edges up to the root edge. The first step of processing the middle sets is to initialize the leaves with values $(0, 0)$, $(1_{[}, 1_{]})$, and $(2_{[}, 2_{]})$. Then, bottom-up, update every pair of states of two child edges e_L and e_R to a state of the parent edge e_P assigning a finite value W_P if the state corresponds to a feasible partial Eulerian labeling.

Let O_L, O_R, and O_P be the nooses corresponding to edges e_L, e_R and e_P. Due to the definition of branch decompositions, every vertex must appear in at least two of the three middle sets and we can define the following partition of the set $(O_L \cup O_R \cup O_P) \cap V(G)$ into sets $I := O_L \cap O_R \cap V(G)$ and $D := O_P \cap V(G) \setminus I$ (I stands for 'Intersection' and D for 'symmetric Difference'). The disc Δ_P bounded by O_P and including the subgraph rooted at e_P contains the union of the discs Δ_L and Δ_R bounded by O_L and O_R and including the subgraphs rooted at e_L and e_R. Thus $|O_L \cap O_R \cap O_P \cap V(G)| \leq 2$. We call the vertices of $O_L \cap O_R \cap O_P \cap V(G)$ *portal vertices*. See the lower part Figure 6 in Appendix for an illustration of these notions.

We compute all valid assignments to the variables $\boldsymbol{t}_P = (v_1, v_2, \ldots, v_p)$ corresponding to the vertices mid(e_P) from all possible valid assignments to the variables of $\boldsymbol{t}_L$ and $\boldsymbol{t}_R$. We use the following notation, for a symbol $x \in \{0, 1_{[}, 1_{\square}, 1_{]}, 2_{[}, 2_{\square}, 2_{]}\}$ we denote by $|x|$ its 'numerical' part. Thus for example, $|1_{[}| = 1$. We say that an assignment c_P is *formed* by assignments c_L and c_R if for every vertex $v \in (O_L \cup O_R \cup O_P) \cap V(G)$:

1. $\underline{v \in D}$: $c_P(v) = c_L(v)$ if $v \in O_L \cap V(G)$, or $c_P(v) = c_R(v)$ otherwise.
2. $\underline{v \in I \setminus O_P}$: $(|c_L(v)| + |c_R(v)|) \, mod \, 2 = 0$ and $|c_L(v)| + |c_R(v)| > 0$.
3. $\underline{v \text{ portal vertex}}$: $|c_P(v)| = 0$ if $|c_L(v)| + |c_R(v)| = 0$, $|c_P(v)| = 1$ if $(|c_L(v)| + |c_R(v)|) \, mod \, 2 = 1$, else $|c_P(v)| = 2$.

Note that for a vertex $v \in O_P \cap V(G)$ it is possible that $|c_P(v)| = 0$ even if $|c_L(v)| + |c_R(v)|$ is even and positive since v might be the only intersection of a component with O_P. In order to verify that the encoding formed from two states of e_L and e_R corresponds to a labeling with each component touching O_P, we use an auxiliary graph A with $V(A) = (O_L \cup O_R) \cap V(G)$ and $\{v, w\} \in E(A)$ if v and w both are in one component of $G_{\mathcal{P}[e_L]}$ and $G_{\mathcal{P}[e_R]}$, respectively. Every component of A must have a vertex in $O_P \cap V(G)$. We compute all ℓ-tuples for $\mathrm{mid}(e_P)$ that can be formed by tuples corresponding to $\mathrm{mid}(e_L)$ and $\mathrm{mid}(e_R)$ and by using the auxiliary graph check if the obtained assignment corresponds to a labeling with each component touching O_P. For every encoding of $\boldsymbol{t}_P$, we set $W_P = min\{W_P, W_L + W_R\}$.

For the root edge $\{r, s\}$ and its children e' and e'' note that $(O_{e'} \cup O_{e''}) \cap V(G) = I$ and $O_{\{r,s\}} = \emptyset$. Hence, for every $v \in V(G_{\mathcal{P}[\{r,s\}]})$ it must hold that $deg_{\mathcal{P}[\{r,s\}]}(v)$ is positive and even, and that the auxiliary graph A is connected. The optimal Eulerian labeling of G results from $min_{\boldsymbol{t}_{\{r,s\}}}\{W_r\}$.

Analyzing the algorithm, we obtain the following lemma.

Lemma 1. Planar Graph TSP *on a graph G with branchwidth $\leq \ell$ can be solved in time $O(2^{5.0998\ell}\ell n + n^4)$.*

Proof. By Theorem 1, an sc-branch decomposition T of width $\leq \ell$ of G can be found in $O(n^4)$.

For three adjacent edges e_P, e_L, and e_R of T and the nooses O_L, O_R, the number of states which have to be updated can be calculated in size of the D- and I-sets. Since the values of the vertices in the D-set are simply transferred the time needed is $O(7^{|D \cap O_L|}7^{|D \cap O_R|})$.

For $c_1, c_2 \in \{0, 1_{[}, 1_{]}, 1_{\square}, 2_{[}, 2_{]}, 2_{\square}\}$ there are 24 possible assignments for $|c_1|+|c_2|$ being positive and even. Hence, the updating time for every I-set is $O(24^{|I|})$. Since the sum of two of each $|D \cap O_L|$, $|D \cap O_R|$ and $|I|$ is at most ℓ we have that $7^{|D \cap O_L|}7^{|D \cap O_R|}24^{|I|} \leq 7^{0.5\ell}7^{0.5\ell}24^{0.5\ell}$. Since there are at most two portal vertices, we neglect them in our calculation.

Since the auxiliary graph is constructed in time linear in the size of states and the number of edges in the tree of a branch decomposition is $O(n)$, we obtain an overall running time of $O(7^{|\ell|}24^{|\ell|}\ell n)$. □

By Proposition 1 and Lemma 1 we obtain the following theorem.

Theorem 2. Planar Graph TSP *is solvable in time $O(2^{10.8224\sqrt{n}}n^{3/2} + n^4)$.*

4 Parameterized planar k-path

In this section we show how sc-branch decompositions can be used to design subexponential parameterized algorithms. We demonstrate it on the parameterized version of Planar Hamiltonian Cycle problem, which for a given planar graph G, asks if G contains a cycle of length n: the Planar k-Cycle problem where a parameter k is given and we ask if there is a cycle of length at least k. We first show how the method works for Planar Hamiltonian Cycle, then we extend it for Planar Longest Cycle (which asks for the longest cycle in a planar graph visiting vertices only once) and only then proceed with the parameterized algorithm. The algorithm for Planar Hamiltonian Cycle is very similar to the algorithm for Planar Graph TSP so we sketch here only the most important details.

Observe that every subgraph partitioned by a noose induces *vertex-disjoint paths* $P_1, \ldots, P_q$ of an Hamiltonian cycle—i.e., every vertex of a subgraph is in at most one path which is part of the entire cycle.

We describe dynamic programming along an sc-branch decomposition $\langle T, \mu, \pi \rangle$ with the same notions as in the previous section with subgraph G_e, noose O_e, state ℓ-tuple $\boldsymbol{t}_e$ with one of the four values: 0, $1_{[}$, $1_{]}$, 2. Hence, there are at most $O(4^{\ell}|V(G)|)$

states. The Boolean value $B_e(i_1, \ldots, i_\ell)$ is TRUE iff there are vertex disjoint paths $P_1, \ldots, P_q$ in G_e such that:

1. Every vertex of $V(G_e) \setminus O_e$ is contained in one of the paths P_i, $1 \leq i \leq q$.
2. Every P_i has both its endpoints in $O_e \cap V(G)$; the first endpoint (in ordering π) is represented by $1_[$ and the second by $1_]$.
3. Every vertex v of $O_e \cap V(G)$ represented by 2 is an inner vertex of one of the paths P_i.
4. Every vertex of $O_e \cap V(G)$ represented by 0 is not contained in any P_i.

If a vertex is represented by 0 then it is adjacent to two edges of a path in the subgraph induced by the leaves of the residual part of e. Since none of $P_1, \ldots, P_q$ cross, the *algebraic term* arguments hold for encoding $1_[$ and $1_]$. A vertex v represented by 2 already is adjacent to two edges of a path of $P_1, \ldots, P_q$. Though v is not clearly encoded, we note that it is superfluously to memorize explicitly to which of $P_1, \ldots, P_q$ vertex v belongs to.

The leaf-edges are initialized by $(0\ 0)$ and $(1_[\ 1_])$. Again, update bottom-up every pair of states of two child edges to get an encoding of the state of the parent edge e_P with B_P is TRUE if every vertex in $V(G_P) \setminus O_P$ belongs to one path with both endpoints in $O_P \cap V(G)$. With the assignments c_e for the vertices in mid(e), we differentiate between the D-,I-sets and portal vertices:

1. $\underline{v \in D}$: $c_P(v) = c_L(v)$ if $v \in O_L \cap V(G)$, or $c_P(v) = c_R(v)$ otherwise.
2. $\underline{v \in I \setminus O_P}$: $(|c_L(v)| + |c_R(v)|) = 2$.
3. $\underline{v \text{ portal vertex}}$: $|c_P(v)| = |c_L(v)| + |c_R(v)| \leq 2$.

We construct auxiliary graph A with $V(A) = (O_L \cup O_R) \cap V(G)$ and $\{v, w\} \in E(A)$ if v and w the endpoints of one path in G_L and G_R, respectively, and check that A contains no cycle. For the root edge e_r, every vertex of G_r must be an inner vertex of a path and A is a cycle.

Lemma 2. PLANAR HAMILTONIAN CYCLE *on a graph G with branchwidth ℓ can be solved in time* $O(2^{3.292\ell}\ell n + n^4)$.

Proof. Using the same notions as in the proof of Lemma 1, the time needed for updating the D-set is $O(4^{|D\cap O_L|}4^{|D\cap O_R|})$. For $c_1, c_2 \in \{0, 1_[, 1_], 2\}$ there are six assignments such that $|c_1| + |c_2| = 2$, and hence, the updating time for every I-set is $O(6^{|I|})$. As in the proof of Lemma 1, we conclude that the worst running time $O(4^{|D\cap O_L|}4^{|D\cap O_R|}6^{|I|})$ occurs for $|D \cap O_L| = |D \cap O_R| = |I| = 0.5\ell$.

□

Theorem 3. PLANAR HAMILTONIAN CYCLE *is solvable in time* $O(2^{6.982\sqrt{n}}n^{3/2} + n^4)$.

Longest Cycle on Planar Graphs.

Let C be a cycle in G. For an edge e of sc-branch decomposition tree T, the noose O_e can affect C in two ways: Either cycle C is partitioned by O_e such that in G_e the remains of C are vertex-disjoint paths , or C is not touched by O_e and thus is completely in G_e or $G \setminus E(G_e)$.

With the same encoding as for PLANAR HAMILTONIAN CYCLE, we add a counter for all states $\boldsymbol{t}_e$ which is initialized by 0 and counts the maximum number of edges over all possible vertex-disjoint paths represented by one $\boldsymbol{t}_e$. In contrast to PLANAR HAMILTONIAN CYCLE, we allow for every vertex $v \in I$ that $|c_L(v)| + |c_R(v)| = 0$ in order to represent the isolated vertices. In the auxiliary graph A, a cycle as a connected component is allowed if all other components are isolated vertices. Then all other vertices in $V(G) \setminus V(G_P)$ of the residual part of T must be of value 0.

Implementing a counter Z for the actual longest cycle, a state in $\boldsymbol{t}_P$ consisting of only 0's represents a collection of isolated vertices with Z storing the longest path in G_P without vertices in $\mathrm{mid}(e)$.

At the root edge, Z gives the size of the longest cycle.

Lemma 3. Planar Longest Cycle *on a graph G with branchwidth ℓ can be solved in time $O(2^{3.4\ell}\ell n + n^4)$.*

Theorem 4. Planar Longest Cycle *is solvable in time* $O(2^{7.214\sqrt{n}}n^{3/2} + n^4)$.

k-Cycle on Planar graphs. Lemma 3 can be used for obtaining parameterized algorithms by adopting the techniques from [8,11].

Before we proceed, let us remind the notion of a minor. A graph H obtained by a sequence of edge-contractions from a graph G is said to be a *contraction* of G. H is a *minor* of G if H is the subgraph of a some contraction of G. Let us note that if a graph H is a minor of G and G contains a cycle of length $\geq k$, then so does G.

We need the following combination of statements (4.3) in [19] and (6.3) in [18].

Theorem 5 ([18]). *Let $k \geq 1$ be an integer. Every planar graph with no $(k \times k)$-grid as a minor has branchwidth $\leq 4k - 3$.*

It easy to check that every $(\sqrt{k} \times \sqrt{k})$-grid, $k \geq 2$, contains a cycle of length $\geq k - 1$. This observation combined with Theorem 5 suggests the following parameterized algorithm. Given a planar graph G and integer k, first compute the branchwidth of G. If the branchwidth of G is at least $4\sqrt{k+1} - 3$ then by Theorem 5, G contains $(\sqrt{k+1} \times \sqrt{k+1})$-grid as a minor and thus contains a cycle of length $\geq k$. If the branchwidth of G is $< 4\sqrt{k+1} - 3$, by Lemma 3, we can find the longest cycle in G in time $O(2^{13.6\sqrt{k}}\sqrt{k}\, n + n^4)$. We conclude with the following theorem.

Theorem 6. Planar k-Cycle *is solvable in time* $O(2^{13.6\sqrt{k}}\sqrt{k}\, n + n^4)$.

By standard techniques (see for example [9]) the recognition algorithm for Planar k-Cycle can easily be turned into a constructive one.

5 Conclusive Remarks

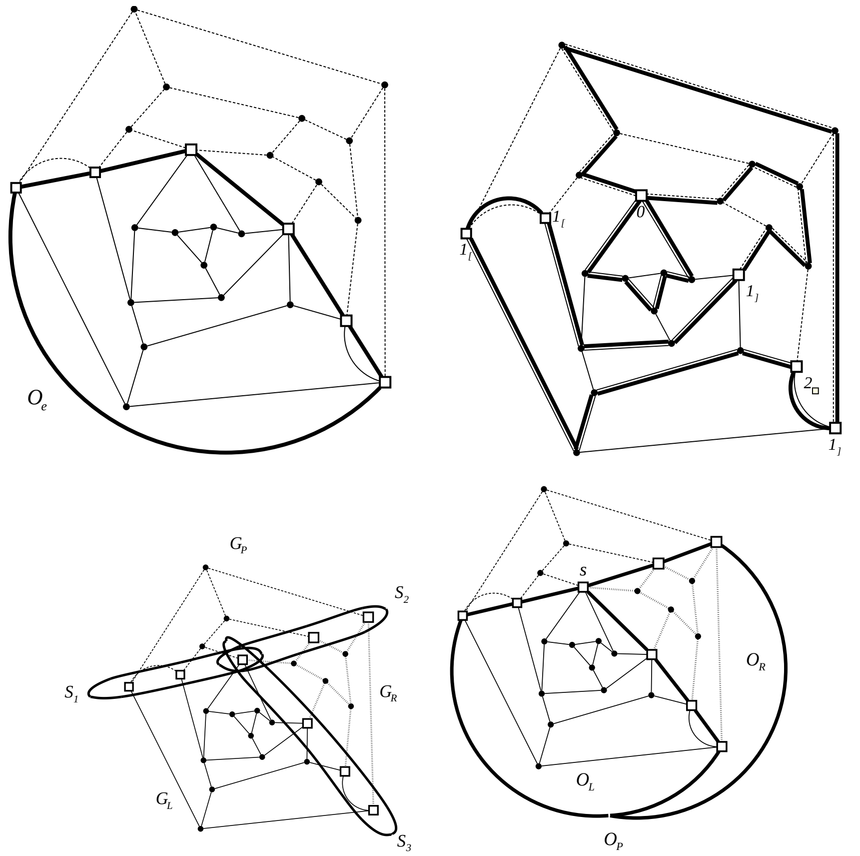
In this paper we introduce a new algorithm design technique based on geometric properties of branch decompositions. Our technique can be also applied to constructing $2^{O(\sqrt{n})} \cdot n^{O(1)}$-time algorithms for a variety of cycle, path, or tree subgraph problems in planar graphs like Hamiltonian Path, Longest Path, and Steiner Tree among others. An interesting question here is if the technique can be extended to more general problems, like Subgraph Isomorphism. For example, Eppstein [10] showed that Planar Subgraph Isomorphism problem with pattern of size k can be solved in time $2^{O(\sqrt{k}\log k)}n$. Can we get rid of the logarithmic factor in the exponent (maybe in exchange to a higher polynomial degree)?

The results of Cook & Seymour [6] on using branch decomposition to obtain high-quality tours for (general) TSP show that branch decomposition based algorithms work much faster than their worst case time analysis shows. Together with our preliminary experience on the implementation of a similar algorithm technique for solving Planar Vertex Cover in [2], we conjecture that sc-branch decomposition based algorithms perform much faster in practice.

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 (2002), pp. 461–493.
2. J. Alber, F. Dorn, and R. Niedermeier, *Experimental evaluation of a tree decomposition-based algorithm for vertex cover on planar graphs*, Discrete Applied Mathematics, 145 (2005), pp. 219–231.
3. J. Alber, H. Fernau, and R. Niedermeier, *Graph separators: a parameterized view*, Journal of Computer and System Sciences, 67 (2003), pp. 808–832.
4. ———, *Parameterized complexity: exponential speed-up for planar graph problems*, Journal of Algorithms, 52 (2004), pp. 26–56.
5. H. L. Bodlaender, *A tourist guide through treewidth*, Acta Cybernet., 11 (1993), pp. 1–21.
6. W. Cook and P. Seymour, *Tour merging via branch-decomposition*, INFORMS Journal on Computing, 15 (2003), pp. 233–248.
7. V. G. Deĭneko, B. Klinz, and G. J. Woeginger, *Exact algorithms for the Hamiltonian cycle problem in planar graphs*, Oper. Res. Lett., (2005), p. to appear.
8. E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, in 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), New York, 2004, ACM, pp. 823–832.
9. R. G. Downey and M. R. Fellows, *Parameterized complexity*, Springer-Verlag, New York, 1999.
10. D. Eppstein, *Subgraph isomorphism in planar graphs and related problems*, Journal of Graph Algorithms and Applications, 3 (1999), pp. 1–27.
11. F. Fomin and D. Thilikos, *Dominating sets in planar graphs: Branch-width and exponential speed-up*, in 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), New York, 2003, ACM, pp. 168–177.
12. ———, *A simple and fast approach for solving problems on planar graphs*, in 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS), vol. 2996 of Lecture Notes in Comput. Sci., Berlin, 2004, Springer, pp. 56–67.
13. M. Held and R. M. Karp, *A dynamic programming approach to sequencing problems*, Journal of SIAM, 10 (1962), pp. 196–210.
14. R. Z. Hwang, R. C. Chang, and R. C. T. Lee, *The searching over separators strategy to solve some NP-hard problems in subexponential time*, Algorithmica, 9 (1993), pp. 398–423.
15. E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, eds., *The traveling salesman problem*, John Wiley & Sons Ltd., Chichester, 1985.
16. R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM Journal on Applied Mathematics, 36 (1979), pp. 177–189.
17. ———, *Applications of a planar separator theorem*, SIAM Journal on Computing, 9 (1980), pp. 615–627.
18. N. Robertson, P. Seymour, and R. Thomas, *Quickly excluding a planar graph*, Journal of Combinatorial Theory Series B, 62 (1994), pp. 323–348.
19. N. Robertson and P. D. Seymour, *Graph minors. X. Obstructions to tree-decomposition*, Journal of Combinatorial Theory Series B, 52 (1991), pp. 153–190.
20. P. Seymour and R. Thomas, *Call routing and the ratcatcher*, Combinatorica, 15 (1994), pp. 217–241.
21. W. D. Smith and N. C. Wormald, *Geometric separator theorems and applications*, in The 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998), IEEE Computer Society, 1998, pp. 232–243.

6 Appendix



On the upper left we see a plane graph G— 3-connected and non-Hamiltonian— partitioned by the rectangle vertices hit by the marked noose O_e into G_e in drawn-through edges and $\overline{G_e}$ in dashed edges. To the right a subgraph $G_{\mathcal{E}}$ with Eulerian labeling $\mathcal{E}$ is marked. $G_{\mathcal{E}}$ is partitioned by the vertices of $O_e \cap V(G)$ which are labeled corresponding to partial Eulerian labeling $\mathcal{P}[e]$ of G_e. Encoding the vertices touched by O_e from the left to the right with $1_[, 1_[, 0, 1_], 2_\square, 1_]$, $G_{\mathcal{P}[e]}$ consists of three components C_1, C_2 and C_3 with $C_1 \cap O_e = \{1_[, 2_\square, 1_]\}, C_2 \cap O_e = \{0\}, C_3 \cap O_e = \{1_[, 1_]\}$. Here $G_{\mathcal{P}[e]}$ has edges only labeled with 1. On the lower left we see the same graph G as in the last figure. G is partitioned by the rectangle vertices of S_1, S_2, S_3 into G_L in drawn-through edges, G_R in dashed edges, and G_P in pointed edges. To the right the three nooses O_L, O_R and O_P are marked. The nooses are induced by S_1, S_2, S_3. Here, S_3 equals the I-set and $S_1 \cup S_2 \setminus I = D$. All three nooses here intersect in one portal vertex s.