REPORTS
IN
INFORMATICS

ISSN 0333-3590

Efficient Exact Algorithms on Planar Graphs: Exploiting Sphere Cut Branch Decompositions

Frederic Dorn and Eelko Penninkx and Hans L. Bodlaender and Fedor V. Fomin

REPORT NO 303 **July 2005**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2005-303.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

Efficient Exact Algorithms on Planar Graphs: Exploiting Sphere Cut Branch Decompositions ⋆

Frederic Dorn[1], Eelko Penninkx[2], Hans L. Bodlaender[2], and Fedor V. Fomin[1]

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

[2] Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands, epennink@cs.uu.nl, hansb@cs.uu.nl

Abstract Divide-and-conquer strategy based on variations of the Lipton-Tarjan planar separator theorem has been 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 based on properties of non-crossing partitions. Compared to 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 $O(2^{6.903\sqrt{n}}n^{3/2}+n^3)$ time algorithm solving weighted Hamiltonian Cycle. We observe how our technique can be used to solve Planar Graph TSP in time $O(2^{10.8224\sqrt{n}}n^{3/2}+n^3)$. 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 $O(2^{13.6\sqrt{k}}\sqrt{k}\,n+n^3)$.

1 Introduction

The celebrated Lipton-Tarjan planar separator theorem [18] is one of the basic approaches to obtain algorithms with subexponential running time for many problems on planar graphs [19]. 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 a serious obstacle for practical implementation. 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 $O(n^3)$ [22] and [13]. Putting all together, we obtain an $O(2^{3.182\sqrt{n}}n+n^3)$ time algorithm solving Independent Set on planar graphs. Note that planarity comes into play twice in this approach: First in the upper bound

⋆ This work was partially supported by Norges forskningsråd project 160778/V30, and partially by the Netherlands Organisation for Scientific Research NWO (project *Treewidth and Combinatorial Optimisation*).

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 from Graph Minors [20] works for many parameterized problems on planar graphs [8]. Using such an approach to solve, for example, Hamiltonian cycle we end up with an $2^{O(\sqrt{n}\log n)}n^{O(1)}$ algorithm on planar graphs, as all known algorithms for this problem on graphs of treewidth ℓ require $2^{O(\ell \log \ell)}n^{O(1)}$ steps. 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: for the first time planarity is used in dynamic programming on graphs of bounded branchwidth.

Our results are based on deep results of Seymour & Thomas [22] on geometric properties of planar branch decompositions. Loosely speaking, their results 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. A refined combinatorial analysis of the algorithm shows that the running time can be calculated by the number of combinations of non-crossing partitions. To demonstrate the power of the new method we apply it to the following problems.

<u>Planar Hamiltonian Cycle</u>. 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 [17]. One of the first known exact exponential time algorithms is the algorithm of Held and Harp [14] 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 (where the cites are points in the Euclidean plane and the distances between the cites are Euclidean distances), several researchers independently obtained subexponential algorithms of running time $2^{O(\sqrt{n}\cdot logn)}n^{O(1)}$ by exploiting planar separator structures (see e.g. [15]). Smith & Wormald [23] 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 roughly $2^{126\sqrt{n}}n^{O(1)}$ for this problem. Because their goal was to get rid of the logarithmic factor in the exponent, they put no efforts in optimizing their algorithm. But even with careful analysis, it is difficult to obtain small constants in the exponent of the divide-and-conquer algorithm due to its recursive nature.

In this paper we use sphere cut branch decompositions not only to obtain a $O(2^{6.903\sqrt{n}}n^{3/2} + n^3)$ time algorithm for Planar Hamiltonian Cycle, but also the first $2^{O(\sqrt{n})}n^{O(1)}$ time algorithm for a generalization, Planar Graph TSP, which for a given weighted planar graph G is a TSP with distance metric the shortest path metric of G.

<u>Parameterized Planar k-cycle</u>. 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. $f(k) = 2^{O(k)}$. During the last two years much attention was paid to the construction of algorithms with running time $2^{O(\sqrt{k})}n^{O(1)}$ 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 an $O(2^{13.6\sqrt{k}}k\,n+n^3)$ 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 decompositions, 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 G only in vertices. The length of a noose O is $|O\cap V(G)|$, 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. all internal vertices of degree three) tree T and a bijection $\mu: L\to E(G)$ between the set L of 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,* $\mathrm{bw}(G)\le\sqrt{4.5n}\le 2.122\sqrt{n}$.

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 there exists a noose O_e bounding the two open discs Δ_1 and Δ_2 such that $G_i\subseteq\Delta_i\cup O_e$, $1\le i\le 2$. Thus O_e meets G only in $\mathrm{mid}(e)$ and its length is $|\,\mathrm{mid}(e)|$. Clockwise traversal of O_e in the drawing of G defines the cyclic ordering π of $\mathrm{mid}(e)$. We always assume that 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 [22] and Gu & Tamaki [13]. Since this result is not explicitly mentioned in [22], 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 $O(n^3)$.*

Proof. Let G be a Σ-plane graph of branchwidth $\leq \ell$ and with minimal vertex degree ≥ 2. Then, $I(G)$ is the simple bipartite graph with vertex $V(G) \cup E(G)$, in which $v \in V(G)$ is adjacent to $e \in E(G)$ if and only if v is an end of e in G. 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 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 $I(G)$ 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 $I(G)$; then C_v has vertex set $\{w_1, \ldots, w_t\}$ and 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 & Thomas ((5.1) and (7.2) [22]) 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 [22]) how to construct an optimal bond carving decompositions of the medial graph M_G in time $O(n^4)$. A refinement of the algorithm in [13] give running time $O(n^3)$. A bond carving decomposition $\langle T, \mu \rangle$ of M_G is also a 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 [22]) 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. □

Non-Crossing Matchings. Together with sphere cut branch decompositions, non-crossing matchings give us the key to our later dynamic programming approach. A *non-crossing partitions (ncp)* is a partition $P(n) = \{P_1, \ldots, P_m\}$ of the set $S = \{1, \ldots, n\}$ such that there are no numbers $a < b < c < d$ where $a, c \in P_i$, and $b, d \in P_j$ with $i \neq j$. A partition can be visualized by a circle with n equidistant vertices on it's border, where every set of the partition is represented by the convex polygon with it's elements as endpoints. A partition is non-crossing if these polygons do not overlap. Non-crossing partitions were introduced by Kreweras [16], who showed that the number of non-crossing partitions over n vertices is equal to the n-th Catalan number:

$$\mathrm{CN}(n) = \frac{1}{n+1}\binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi} n^{\frac{3}{2}}} \approx 4^n \quad (1)$$

A *non-crossing matching (ncm)* is a special case of a ncp, where $|P_i| = 2$ for every element of the partition. A ncm can be visualized by placing n vertices on a straight line, and connecting matching vertices with arcs at one fixed side of the line. A matching is non-crossing if these arcs do not cross. The number of non-crossing matchings over n vertices is given by:

$$\mathrm{M}(n) = \mathrm{CN}(\frac{n}{2}) \sim \frac{2^n}{\sqrt{\pi}(\frac{n}{2})^{\frac{3}{2}}} \approx 2^n \quad (2)$$

3 Planar Hamiltonian Cycle

In this section we show how sc-branch decompositions in combination with ncm's can be used to design subexponential parameterized algorithms. In the Planar Hamiltonian Cycle problem we are given a weighted Σ-plane graph $G = (V, E)$ with weight function $w\colon E(G) \to N$ and we ask for a cycle of minimum weight through all vertices of V. We can formulate the problem in a different way: A labelling $\mathcal{H} : E(G) \to \{0, 1\}$ is *Hamiltonian* if the subgraph $G_{\mathcal{H}}$ of G formed by the edges with positive labels is a spanning cycle. Find a Hamiltonian labelling $\mathcal{H}$ minimizing $\sum_{e \in E(G)} \mathcal{H}(e) \cdot w(e)$. For an edge labelling $\mathcal{H}$ and vertex $v \in V(G)$ we define the *$\mathcal{H}$-degree* $deg_{\mathcal{H}}(v)$ of v as the sum of labels assigned to the edges incident to v. Though the use of labellings makes the algorithm more sophisticated, it is necessary for the understanding of the latter approach for Planar Graph TSP. Let $\langle T, \mu, \pi \rangle$ be a sc-branch decomposition of G of width ℓ. We root T by arbitrarily choosing an edge e, and subdivide it by inserting a new node s. Let e', e'' be the new edges and set $\mathrm{mid}(e') = \mathrm{mid}(e'') = \mathrm{mid}(e)$. Create a new node *root* r, connect it to s and set $\mathrm{mid}(\{r, s\}) = \emptyset$. Each internal 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 the subtree towards the leaves is called the *lower part* and the rest the *residual part* with regard 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. The noose O_e partitions Σ into two discs, one of which, Δ_e, contains G_e.

We call a labelling $\mathcal{P}[e]\colon E(G_e) \to \{0, 1\}$ a *partial Hamiltonian labelling* if the subgraph $G_{\mathcal{P}[e]}$ induced by the edges with positive labels satisfies the following properties:

- For every vertex $v \in V(G_e) \setminus O_e$, the $\mathcal{P}[e]$-degree $deg_{\mathcal{P}[e]}(v)$, i.e. the sum of labels assigned by $\mathcal{P}[e]$ to the edges incident to v, is two.
- Every connected component of $G_{\mathcal{P}[e]}$ has two vertices in O_e with $deg_{\mathcal{P}[e]}(v) = 1$ for $e \neq \{r, s\}$; For $e = \{r, s\}$, $G_{\mathcal{P}[e]}$ is a cycle.

Observe that $G_{\mathcal{P}[e]}$ forms a collection of disjoint paths, and note that every partial Hamiltonian labelling of $G_{\{r,s\}}$ is also a Hamiltonian labelling.

For dynamic programming we need to keep for every edge e of T the information on which vertices of the disjoint paths of $G_{\mathcal{P}[e]}$ of all possible partial Hamiltonian labellings $\mathcal{P}[e]$ 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 1, or 2.

And here the geometric properties of sc-branch decompositions in combination with ncm's come into play. For a partial Hamiltonian labelling $\mathcal{P}[e]$ let P be a path of $G_{\mathcal{P}[e]}$. We scan the vertices of $V(P) \cap O_e$ according to the ordering π and mark with '$1_[$' the first and with '$1_]$' the last vertex of P on O_e. We also mark by '2' the other 'inner' vertices of $V(P) \cap O_e$. 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 '$1_{[}$', '$1_{]}$', '2', and '0', one can decode the complete information on which vertices of each path of $V(G_{\mathcal{P}[e]})$ hit O_e. This is possible because O_e bounds the disc Δ_e and the graph $G_{\mathcal{P}[e]}$ is in Δ_e. The sets of endpoints of every path are the elements of an ncm. Hence, with the given ordering π the '$1_{[}$' and '$1_{]}$' encode an ncm.

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. This order is necessary for a well-defined encoding for the states when allowing $v_1, \ldots, v_\ell$ to have one of the four values: 0, $1_{[}$, $1_{]}$, 2. Hence, there are at most $O(4^\ell |V(G)|)$ 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 Hamiltonian labelling $\mathcal{P}[e]$ such that:

1. For every path P of $G_{\mathcal{P}[e]}$ the first vertex of $P \cap O_e$ in π is represented by $1_{[}$ and the last vertex is represented by $1_{]}$. All other vertices of $P \cap O_e$ are represented by 2.
2. Every vertex $v \in (V(G_e) \cap O_e) \setminus G_{\mathcal{P}[e]}$ is marked by 0.

We put $W = +\infty$ if no such labelling exists. For every vertex v the numerical part of its value gives $deg_{\mathcal{P}[e]}(v)$.

To compute an optimal Hamiltonian labelling we perform dynamic programming over middle sets $\text{mid}(e) = O(e) \cap V(G)$, starting at the leaves of T and working bottom-up towards the root edge. The first step in processing the middle sets is to initialize the leaves with values $(0, 0)$, $(1_{[}, 1_{]})$. 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 Hamiltonian labelling.

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 $|O_L \cap O_R \cap O_P \cap V(G)| \leq 2$. The vertices of $O_L \cap O_R \cap O_P \cap V(G)$ are called *portal vertices*.

We compute all valid assignments to the variables $\boldsymbol{t}_P = (v_1, v_2, \ldots, v_p)$ corresponding to the vertices $\text{mid}(e_P)$ from all possible valid assignments to the variables of $\boldsymbol{t}_L$ and $\boldsymbol{t}_R$. For a symbol $x \in \{0, 1_{[}, 1_{]}, 2\}$ we denote by $|x|$ its 'numerical' part. 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. $\underline{v \in I \setminus O_P}$: $(|c_L(v)| + |c_R(v)|) = 2$.
3. $\underline{v \text{ portal vertex}}$: $|c_P(v)| = |c_L(v)| + |c_R(v)| \leq 2$.

We compute all ℓ-tuples for $\text{mid}(e_P)$ that can be formed by tuples corresponding to $\text{mid}(e_L)$ and $\text{mid}(e_R)$ and check if the obtained assignment corresponds to a labelling without cycles. 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 two, and that the labellings form a cycle. The optimal Hamiltonian labelling of G results from $min_{\boldsymbol{t}_{\{r,s\}}}\{W_r\}$.

Analyzing the algorithm, we obtain the following lemma.

Lemma 1. Planar Hamiltonian Cycle *on a graph G with branchwidth ℓ can be solved in time $O(2^{3.292\ell}\ell n + n^3)$.*

Proof. By Theorem 1, an sc-branch decomposition T of width $\leq \ell$ of G can be found in $O(n^3)$.

Assume we have three adjacent edges e_P, e_L, and e_R of T with $|O_L| = |O_R| = |O_P| = \ell$. Without loss of generality we limit our analysis to even values for ℓ, and for simplicity assume there are no portal vertices. This can only occur if $|I| = |D \cap O_L| = |D \cap O_R| = \frac{\ell}{2}$.

By just checking every combination of ℓ-tuples from O_L and O_R we obtain a bound of $O(\ell 4^{2\ell})$ for our algorithm.

Some further improvement is apparent, as for the vertices $u \in I$ we want the sum of the $\{0, 1_[, 1_], 2\}$ assignments from both sides to be 2.

We start by giving an expression for $Q(\ell, m)$: the number of ℓ-tuples over ℓ vertices where the $\{0, 1_[, 1_], 2\}$ assignments for vertices from I is fixed and contains m $1_[$'s and $1_]$'s. The only freedom is thus in the $\ell/2$ vertices in $D \cap O_L$ and $D \cap O_R$, respectively:

$$\mathrm{Q}(\ell, m) = \sum_{i=m\%2}^{\frac{\ell}{2},2} \binom{\frac{\ell}{2}}{i} 2^{\frac{\ell}{2}-i} \mathrm{CN}(\frac{i+m}{2}) \tag{3}$$

This expression is a summation over the number of $1_[$'s and $1_]$'s in $D \cap O_L$ and $D \cap O_R$, respectively. The starting point is $m\%2$ (m modulo 2), and the 2 at the top of the summation indicates that we increase i with steps of 2, as we want $i + m$ to be even (the $1_[$'s and $1_]$'s have to form a ncm). The term $\binom{\frac{\ell}{2}}{i}$ counts the possible locations for the $1_[$'s and $1_]$'s, the $2^{\frac{\ell}{2}-i}$ counts the assignment of $\{0, 2\}$ to the remaining $\ell/2 - i$ vertices, and the $\mathrm{CN}(\frac{i+m}{2})$ term counts the ncm's over the $1_[$'s and $1_]$'s. As we are interested in exponential behaviour for large values of ℓ we ignore that $i + m$ is even, and use that $CN(n) \approx 4^n$:

$$Q(\ell, m) \approx \sum_{i=0}^{\frac{\ell}{2}} \binom{\frac{\ell}{2}}{i} 2^{\frac{\ell}{2}-i} 2^{i+m} = 2^{\frac{\ell}{2}+m} \sum_{i=0}^{\frac{\ell}{2}} \binom{\frac{\ell}{2}}{i} = 2^{\ell+m} \tag{4}$$

We can now count the total cost of forming an ℓ-tuple from O_P. We sum over i: the number of $1_[$'s and $1_]$'s in the assignment for I:

$$C(\ell) = \sum_{i=0}^{\frac{\ell}{2}} \binom{\frac{\ell}{2}}{i} 2^{\frac{\ell}{2}-i} Q(\ell, i)^2 \tag{5}$$

Straightforward calculation by approximation yields:

$$C(\ell) \approx \sum_{i=0}^{\frac{\ell}{2}} \binom{\frac{\ell}{2}}{i} 2^{\frac{\ell}{2}-i} 2^{2\ell+2i} = 2^{\frac{5\ell}{2}} \sum_{i=0}^{\frac{\ell}{2}} \binom{\frac{\ell}{2}}{i} 2^i = 2^{\frac{5\ell}{2}} 3^{\frac{\ell}{2}} = (4\sqrt{6})^\ell \tag{6}$$

By Proposition 1 and Lemma 1 we achieve the running time $O(2^{6.987\sqrt{n}} n^{3/2} + n^3)$ for Planar Hamiltonian Cycle.

3.1 Forbidding Cycles

We can further improve upon the previous bound by only forming encodings that don't create a cycle. As cycles can only be formed at the vertices in I with numerical part 1 in both O_L and O_R, we only consider these vertices. Note that within these vertices both in O_L and O_R every vertex with a $1_]$ has to be paired with a $1_[$, whereas a $1_[$ could be paired with a $1_]$ that lies in D. We encode every vertex by a $\{1_[, 1_]\}^2$ assignment, where the first corresponds to the state in O_L, and the second to the state in O_R. For example $|1_[1_[, 1_]1_]|$ corresponds to a cycle over two vertices. We obtain the following combinatorial problem: given n vertices with a $\{1_[, 1_]\}^2$ assignment to every vertex, how many combinations can be formed that induce no cycles and pair every $1_]$ with a $1_[$ at the same side?

Exact counting is complex, so we use a different approach. We try to find some small z such that $|B(b)|$ is $O(z^n)$. Let $B(i)$ denote the set of all feasible combinations over i vertices: $B(0) = \emptyset$, $B(1) = \{|1_[1_[|\}$, $B(2) = \{|1_[1_[, 1_[1_[|, |1_[1_[, 1_]1_[|, |1_[1_[, 1_[1_]|\}$, etc. Note that $|1_[1_[, 1_]1_]|$ is not included in $B(2)$ as this is a cycle. We map all items of $B(i)$ to a fixed number of classes $C_1, \ldots, C_m$ and define $x_i = \{x_{i1}, \ldots, x_{im}\}^T$ as the number of elements in each class.

Suppose we use two classes: C_1 contains all items $|\ldots, 1_[1_[|$, and C_2 contains all other items. Note that adding $1_]1_]$ to items from C_1 is forbidden, as this will lead to a cycle. Addition of $1_[1_[$ to items from C_2 gives us items of class C_1. Addition of $1_[1_]$ or $1_]1_[$ to either class leads to items of class C_2, or can lead to infeasible encodings. These observations show that $A = \begin{pmatrix} 1 & 1 \\ 2 & 3 \end{pmatrix}$. As the largest real eigenvalue of A is $2 + \sqrt{3}$, we have $z \leq 3.73205$.

Using these two classes eliminated all cycles over two consecutive vertices. By using more classes we can prevent larger cycles, and obtain tighter bounds for z. With only three classes we obtain $z \leq 3.68133$. This bound is definitely not tight, but computational research suggests that z is probably larger than 3.5. By using the value 3.68133 for z we obtain the following result:

Theorem 2. Planar Hamiltonian Cycle *is solvable in* $O(2^{6.903\sqrt{n}} n^{3/2} + n^3)$.

4 Variants

In this section we will discuss results on other non-local problems on planar graphs.

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 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. 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 mid(e). At the root edge, Z gives the size of the longest cycle. Thus, Planar Longest Cycle is solvable in time $O(2^{7.223\sqrt{n}} n^{3/2} + n^3)$.

k-Cycle on Planar graphs is the problem of finding a cycle of length at least a parameter k. The algorithm on Longest Cycle 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 [21] and (6.3) in [20].

Theorem 3 ([20]). *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 3 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 3, G contains a $(\sqrt{k+1} \times \sqrt{k+1})$-grid as a minor and thus contains a cycle of length $\geq k$. If the branchwidth of G is $< 4\sqrt{k+1} - 3$ we can find the longest cycle in G in time $O(2^{13.6\sqrt{k}}\sqrt{k}\,n + n^3)$. We conclude with the following theorem.

Theorem 4. PLANAR k-CYCLE *is solvable in time* $O(2^{13.6\sqrt{k}}\sqrt{k}\,n + n^3)$.

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

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. The algorithm for PLANAR GRAPH TSP is very similar to the algorithm for PLANAR HAMILTONIAN CYCLE so we mention here only the most important details. 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. Since every vertex of an Eulerian graph is of even degree we obtain an *Eulerian* labelling $\mathcal{E} : E(G) \to \{0, 1, 2\}$ with 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 is equivalent to finding an Eulerian labelling $\mathcal{E}$ minimizing $\sum_{e \in E(G)} \mathcal{E}(e) \cdot w(e)$.

In contrast to the approach for PLANAR HAMILTONIAN CYCLE, the parity plays an additional role in dynamic programming, and we obtain a bit more sophisticated approach.

Theorem 5. PLANAR GRAPH TSP *is solvable in time* $O(2^{10.8224\sqrt{n}} n^{3/2} + n^3)$.

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 CONNECTED DOMINATING SET, 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] Q.-P. Gu and H. Tamaki, *Optimal branch-decomposition of planar graphs in $O(n^3)$*, in 32nd International Colloquium on Automata, Languages and Programming (ICALP), 2005, p. to appear.

[14] M. Held and R. M. Karp, *A dynamic programming approach to sequencing problems*, Journal of SIAM, 10 (1962), pp. 196–210.

[15] 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.

[16] G. Kreweras, *Sur les partition non croisées d'un circle*, Discrete Mathematics, 1 (1972), pp. 333–350.

[17] E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, eds., *The traveling salesman problem*, John Wiley & Sons Ltd., Chichester, 1985.

[18] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM Journal on Applied Mathematics, 36 (1979), pp. 177–189.

[19] ———, *Applications of a planar separator theorem*, SIAM Journal on Computing, 9 (1980), pp. 615–627.

[20] N. Robertson, P. Seymour, and R. Thomas, *Quickly excluding a planar graph*, Journal of Combinatorial Theory Series B, 62 (1994), pp. 323–348.

[21] N. Robertson and P. D. Seymour, *Graph minors. X. Obstructions to tree-decomposition*, Journal of Combinatorial Theory Series B, 52 (1991), pp. 153–190.

[22] P. Seymour and R. Thomas, *Call routing and the ratcatcher*, Combinatorica, 15 (1994), pp. 217–241.

[23] 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.