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, 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, 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].

* 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*).

G.S. Brodal and S. Leonardi (Eds.): ESA 2005, LNCS 3669, pp. 95–106, 2005.

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 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 noncrossing partitions. To demonstrate the power of the new method we apply it to the following problems.

PLANAR HAMILTONIAN CYCLE. 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