Subexponential Parameterized Algorithms for Degree-Constrained Subgraph Problems on Planar Graphs*

Ignasi Sau[†‡] Dimitrios M. Thilikos[§¶]

January 18, 2010

Abstract

We present subexponential parameterized algorithms on planar graphs for a family of problems of the following shape: given a graph, find a connected (induced) subgraph with bounded maximum degree and with maximum number of edges (or vertices). These problems are natural generalisations of the LONGEST PATH problem. Our approach uses bidimensionality theory combined with novel dynamic programming techniques over branch decompositions of the input graph. These techniques can be applied to a more general family of problems that deal with finding connected subgraphs under certain degree constraints.

Keywords: Parameterized complexity, planar graphs, subexponential algorithm, branch decomposition, graph minors, bidimensionality, Catalan structures.

1 Introduction

During the last years a considerable amount of work has been devoted to design subexponential parameterized algorithms for NP-hard optimisation problems on planar graphs and, more generally, on sparse classes of graphs [3–6]. In this article we apply the general approach of [3–6] to a family of problems dealing with finding connected subgraphs under degree constraints. Along the way, we introduce novel dynamic programming techniques over branch decompositions that can be applied to more general classes of problems.

We define the following family of problems for $d \geq 2$.

---

*A conference version of this work appeared in *Proceedings of DIMAP workshop on Algorithmic Graph Theory (AGT), Electronic Notes in Discrete Mathematics, 32:59-66, University of Warwick, U.K., March 2009.*

†Mascotte joint Project of INRIA/CNRS/UNSA, Sophia-Antipolis, France; and Graph Theory and Combinatorics Group, Departament de Matemàtica Aplicada IV, UPC, Barcelona, Spain. E-mail: `ignasi.sau@sophia.inria.fr`

‡Supported by IST FET AEOLUS, PACA region of France, and COST 295-DYNAMO.

§Department of Mathematics, National and Kapodistrian University of Athens, Greece. E-mail: `sedthilk@math.uoa.gr`

¶Supported by the project "Kapodistrias" (AΠ 02839/28.07.2008) of the National and Kapodistrian University of Athens.

Maximum d-Degree-Bounded Connected Subgraph (MDBCS$_d$) **Input:** A graph G and a non-negative integer k. **Question:** Does G contain a connected subgraph H with maximum degree at most d and at least k edges?

The maximization version of MDBCS$_d$ can be seen as a generalisation of the Longest Path or Cycle problem, which is exactly the case $d = 2$. The edge-maximisation version of MDBCS$_d$ is one of the classical NP-hard problems listed in Garey and Johnson's monograph [13, problem GT26], and it has been recently proved that it is not in Apx for any $d \geq 2$ [1]. Without the connectivity constraint, the problem is known to be solvable in polynomial time using matching techniques [17]. When the problem is parameterized by k we denote it by k-MDBCS$_d$. (We refer to [9] for an introduction to parameterized complexity.) Our target is to find $2^{\mathcal{O}(\sqrt{k})} \cdot \mathcal{O}(n)$ step algorithms to solve this problem and its variants when the input is restricted to planar graphs. Section 3 is devoted to obtain combinatorial bounds using bidimensionality theory. Section 4 presents new dynamic programming techniques, that can be applied to general graphs. In Section 5 we see how to speed-up these algorithms when the input is restricted to planar graphs, using Catalan structures. This strategy can be extended to several related problems asking for a maximum connected subgraph satisfying certain degree constraints, as discussed in Section 6. Some open problems are listed in Section 7.

2 Preliminaries

All the graphs considered in this article are simple and undirected. Given a graph G we denote as $V(G)$ and $E(G)$ the vertices and the edges of G respectively. If H is a subgraph of G, we denote it by $H \subseteq G$. Given a a subset $S \subseteq V(G)$, we define $N_G[S]$ to be the set of vertices of $V(G)$ at distance at most 1 from at least one vertex of S. If $S = \{v\}$, we simply use the notation $N_G[v]$. We also define $N_G(v) = N_G[v] - \{v\}$ and $E_G(v) = \{\{v,u\} \mid u \in N_G(v)\}$. The *degree* of a vertex $v \in V$ is defined as $\mathbf{deg}_G(v) = |N_G(v)|$. The *maximum degree* of G is defined as $\Delta(G) = \max_{v \in V(G)} \mathbf{deg}_G(v)$. Let $e = \{x,y\} \in E(G)$. We denote by $G \backslash e$ the graph G' where $G' = (V(G), E(G) - \{e\})$ and we say that G' *occurs from* G *after an edge removal*. We also denote by G/e the graph G' where

$$G' = (V(G) - \{x,y\} \cup \{v_{x,y}\}, E(G) - E_G(x) - E_G(y) \cup \{\{v_{xy}, z\} \mid z \in N_G[\{x,y\}]\}),$$

where $v_{xy} \notin V(G)$ is a new vertex, not in G. In this case we say that G' *occurs from* G *after an edge contraction*. If H occurs from a subgraph of G after a (possibly empty) sequence of edge contractions, we say that H is a *minor* of G, and that G is a *major* of H.

Let G be a graph on n vertices. A *branch decomposition* (T, μ) of a graph G consists of an unrooted ternary tree T (i.e., all internal vertices are of degree three) and a bijection $\mu : L \to E(G)$ from the set L of leaves of T to the edge set of G. We define for every edge e of T the *middle set* $\mathbf{mid}(e) \subseteq V(G)$ 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., $\mathbf{mid}(e) = V(G_1) \cap V(G_2)$. Note that for each $e \in E(T)$, $\mathbf{mid}(e)$ is a separator of G. The *width* of (T, μ) is the maximum order of the middle sets over all edges of T, i.e., $\max\{|\mathbf{mid}(e)|\colon e \in T\}$. An optimal branch decomposition of G is defined by a tree T and a bijection μ which give the minimum width, the *branchwidth*, denoted by $\mathbf{bw}(G)$. Intuitively, branchwidth is a measure of the local connectivity of a graph, i.e., its topological resemblance to a tree. The following fundamental theorem states that square grids serve as obstructions for branchwidth on planar graphs.

Theorem 1 (Robertson, Seymour, and Thomas [19]) *Let $h \geq 1$ be an integer. Every planar graph of branchwidth at least h contains an $(\lfloor h/4 \rfloor \times \lfloor h/4 \rfloor)$-grid as a minor.*

We say that a parameter $\mathbf{p}$ defined on simple undirected graphs is *closed under taking of minors* (or simply *minor closed*) if $G' \preceq G \Rightarrow \mathbf{p}(G') \leq \mathbf{p}(G)$ (here "$\preceq$" denotes the minor relation). A parameter $\mathbf{p}$ is *minor bidimensional* [3] with *density* δ if

- $\mathbf{p}$ is minor closed, and
- for the $(r \times r)$-grid R, $\mathbf{p}(R) = (\delta r)^2 + o((\delta r)^2)$.

Theorem 1 implies the following useful property.

Lemma 1 (Demaine et al. [3]) *If $\mathbf{p}$ is a bidimensional parameter with density δ then for any planar graph G, $\mathbf{bw}(G) \leq \frac{4}{\delta} \cdot \sqrt{\mathbf{p}(G)} + \mathcal{O}(1)$.*

3 Bounds for Branchwidth

In this section we define the following parameter on simple undirected graphs, and we obtain combinatorial bounds for it using bidimensionality theory.

$$\mathbf{medbcs}_d(G) = \max\{|E(H)| \ \mid \ H \subseteq G \ \wedge \ H \text{ is connected } \wedge \ \Delta(H) \leq d\}.$$

Lemma 2 *For any integer $d \geq 1$, the parameter $\mathbf{medbcs}_d$ is minor closed.*

Proof: If G' occurs from G after an edge removal, then clearly $\mathbf{medbcs}_d(G') \leq \mathbf{medbcs}_d(G)$. Let us see that the same holds if G' occurs from G after the contraction of an edge $\{x, y\}$. Indeed, we shall see that given any connected subgraph $H' \subseteq G'$ with $\Delta(H') \leq d$, we can find a connected subgraph $H^* \subseteq G$ with $\Delta(H^*) \leq d$ and $|E(H^*)| \geq |E(H')|$. Let H be the major of H' in G. We can assume that $v_{xy} \in V(H')$, otherwise we set $H^* = H$. We define $N_{xy} = N_H(x) \cap N_H(y)$, $N_{x-y} = N_H(x) - N_{xy} - \{y\}$, and $N_{y-x} = N_H(y) - N_{xy} - \{x\}$. The subgraph H is connected and $|E(H)| \geq |E(H')|$, but the vertices x, y, and those in N_{xy} may have degree $d+1$. Since $\Delta(H') \leq d$, also $|N_{H'}(v_{xy})| = |N_{x-y}| + |N_{y-x}| + |N_{xy}| \leq d$. Suppose w.l.o.g. that $|N_{x-y}| \geq |N_{y-x}|$. We distinguish several cases to define the subgraph H^*: If $|N_{x-y}| = d$, let $H^* = (V(H) - \{y\}, E(H) - \{x, y\})$. Suppose henceforth that $|N_{x-y}| < d$. If $|N_{xy}| = 0$, let $H^* = H$. If $N_{xy} = \{z_1\}$, let $H^* = (V(H), E(H) - \{x, z_1\})$. Finally, if $N_{xy} = \{z_1, \ldots, z_k\}$ for some $k \geq 2$, let $H^* = (V(H), E(H) - \{x, z_1\} - \cup_{i=2}^{k}\{y, z_i\})$. It is easy to check that, in all cases, the subgraph H^* is connected, $\Delta(H^*) \leq d$, and $|E(H^*)| \geq |E(H')|$. ∎

Using Lemmas 2 and 1 we can obtain a combinatorial bound of the parameter $\mathbf{medbcs}_d$ in terms of the branchwidth of the planar graph G.

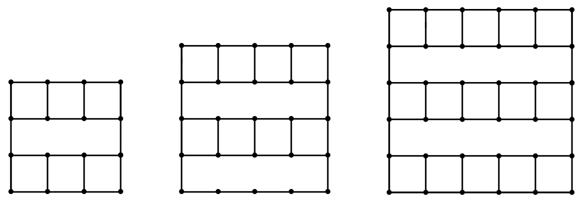

Figure 1: Connected subgraphs with maximum degree 3 on (4×4), (5×5), and (6×6)-grids respectively, used in the proof of Lemma 3.

Lemma 3 *For any $d \geq 2$ and for any planar graph G it holds that*

$$\mathbf{bw}(G) \leq \frac{4}{\delta} \cdot \sqrt{\mathbf{medbcs}_d(G)} + \mathcal{O}(1), \quad \textit{with} \;\; \delta = \begin{cases} 1 & \textit{, if } d = 2 \\ \sqrt{3/2} & \textit{, if } d = 3 \\ \sqrt{2} & \textit{, if } d \geq 4 \end{cases}$$

Proof: We shall prove that the parameter $\mathbf{medbcs}_d(G)$ is bidimensional for any $d \geq 2$. It is minor closed due to Lemma 2. Let us see how the parameter behaves on the grid. Let R be an $(r \times r)$-grid. If $d = 2$, then clearly $\mathbf{medbcs}_2(R) \geq r^2 - 1$ (or r^2 if r is even, because in this case the grid contains a Hamiltonian cycle). That is, $\mathbf{medbcs}_2(R) = r^2 + o(r^2)$, so the density of $\mathbf{medbcs}_2$ is 1. If $d \geq 4$ then the optimal solution contains all the edges, i.e., $\mathbf{medbcs}_d(R) = 2r(r-1) = (\sqrt{2}r)^2 + o((\sqrt{2}r)^2)$. Said otherwise, the density of $\mathbf{medbcs}_d$ for $d \geq 4$ is $\sqrt{2}$. Finally, if $d = 3$, we shall see that $\mathbf{medbcs}_3(R) \geq 2r(r-1) - \lceil \frac{r-2}{2} \rceil (r-2)$. Such a solution is obtained in the following way. Take all the *horizontal* edges of the grid, and the *vertical* edges from the leftmost and rightmost column. Then, beginning from the first row, take alternatively the remaining vertical edges (see Figure 1 for an illustration). One can easily check that the subgraph obtained in this way is connected, has maximum degree 3 and has $2r(r-1) - \lceil \frac{r-2}{2} \rceil (r-2) = 3/2r^2 + o(3/2r^2)$ edges. The coefficient of r^2 is best possible, as the degree of the vertices must be at most 3. That is, $\mathbf{medbcs}_3(R) = (\sqrt{3/2}r)^2 + o((\sqrt{3/2}r)^2)$, so the density of $\mathbf{medbcs}_3$ is $\sqrt{3/2}$. The result follows from Lemma 1. ∎

4 The Algorithms

In this section we present algorithms based on dynamic programming over branch decompositions. It is worth to mention that the methods we use can be directly translated to tree decompositions (see for instance [7]). In addition, it is well-known that the treewidth and the branchwidth of a graph with at least 3 edges differ by a factor at most 3/2 [18]. However, there are several reasons why we chose to work with branch decompositions. First of all, we follow the approach of [3–6, 11], which is based on branch decompositions. Also, an optimal branch decomposition of a planar graph can be constructed in polynomial time [20], whereas the question is still open for tree decompositions. And last, for planar graphs there is a nice type of branch decompositions that allow to speed-up our algorithms (see Section 5).

Roughly speaking, in each edge of the branch decomposition, the tables of our dynamic programming algorithm store all the *partial* solutions to the problem in the graph processed so far. The output subgraph (corresponding to the root) is required to be connected. However, partial solutions may have several connected components, so we need to keep track of them. We also need to control the degrees of the vertices in the partial solutions, in order to assure that the maximum degree of the output subgraph is bounded by d. To do so, we use what we call *weighted packings* of the middle sets (defined below), which encode the connected components and the degrees of the intersection of the partial solutions with the middle set. The tables of each edge are filled from the tables of the two previously processed edges incident to the same vertex, and when two entries are combined, the connected components which intersect are fused and the degrees of the vertices are updated.

Before proceeding to the description of the algorithms, we need some definitions. Let G be in this section a (not necessarily planar) graph on n vertices. We denote the *empty set* by $\emptyset$ and the *empty function* by $\varnothing$. Let (T, μ) be a branch decomposition of width $\leq h$ of G. In order to *root* (T, μ), we pick an arbitrary edge $e^* \in E(T)$, we subdivide it by adding a new vertex v_{new} and then add a new vertex r and make it adjacent to v_{new}. We extend μ by setting $\mu(r) = \emptyset$ and we root T at vertex r. For each $e \in E(T)$ let T_e be the tree of the forest $T \backslash e$ that does not contain r as a leaf (i.e., the tree that is "below" e in the rooted tree T) and let E_e be the edges that are images, via μ, of the leaves of T that are also leaves of T_e. We denote $G_e = G[E_e]$. Observe that, if $e_r = \{v_{\text{new}}, r\}$, then $G_{e_r} = G$.

Given a set A, we define a *d-weighted packing* of A as any pair $(\mathcal{A}, \psi)$ where $\mathcal{A}$ is a (possible empty) collection of mutually disjoint nonempty subsets of A and $\psi : A \to \{0, \ldots, d\}$ is a mapping corresponding integers from 0 to d to the elements of A. It will be convenient to think of such a packing $\mathcal{A}$ of A as a hypergraph $\mathcal{G} = (A, \mathcal{A})$. Note that, by definition, $\mathcal{A}$ is a matching in $\mathcal{G}$. For convenience, given such a collection $\mathcal{A}$, we denote by $\cup\mathcal{A}$ the set $\bigcup_{X \in \mathcal{A}} X$.

Let $(\mathcal{A}, \psi)$ and $(\mathcal{A}', \psi')$ be two d-weighted packings of two sets A and A'. We define $(\mathcal{A}, \psi) \oplus (\mathcal{A}', \psi')$ as the $2d$-weighted packing $(\mathcal{A}'', \psi'')$ of $A'' = A \cup A'$ where $\mathcal{A}''$ is the packing of A'' defined by the connected components of the hypergraph $(A \cup A', \mathcal{A} \cup \mathcal{A}')$ (i.e., the nonempty subsets of the packing $\mathcal{A}''$ are the vertex sets corresponding to the connected components of the hypergraph $(A \cup A', \mathcal{A} \cup \mathcal{A}')$) and where for any $x \in A \cup A'$,

$$\psi''(x) = \begin{cases} \psi(x) & \text{, if } x \in A - A' \\ \psi(x) + \psi'(x) & \text{, if } x \in A \cap A' \\ \psi'(x) & \text{, if } x \in A' - A \end{cases}$$

If $(\mathcal{A}, \psi)$ is a d-weighted packing of a set A and $A' \subseteq A$, we define $(\mathcal{A}, \psi)|_{A'}$ as the d-weighted packing $(\mathcal{A}', \psi')$ of the set A' where $\mathcal{A}' = \{X \cap A' \mid X \in \mathcal{A}\}$ and $\psi' = \{(x, \psi(x)) \mid x \in A'\}$, where $(x, \psi(x)) \in \psi'$ means that $\psi'(x) = \psi(x)$.

Let $\mathscr{P}_e$ be the collection of all d-weighted packings $(\mathcal{A}, \psi)$ of $\mathbf{mid}(e)$, and let $h = |\mathbf{mid}(e)|$. Observe that if $e_r = \{v_{\text{new}}, r\}$, then $\mathscr{P}_{e_r} = \{(\emptyset, \varnothing)\}$. We use the notation $\mathcal{C}(H)$ for the set of connected components of a graph (or hypergraph) H. Given $(\mathcal{A}, \psi) \in \mathscr{P}_e$ we define

$$\begin{aligned}\mathbf{opt}_e(\mathcal{A}, \psi) \;=\; \max\{\{0\} \cup \{|E(H)| \;:\; & H \subseteq G_e \;\wedge\; \Delta(H) \leq d \;\wedge \\ & \text{if } (\mathcal{A} \neq \emptyset) \text{ then} \\ & \quad \{V(H') \cap \mathbf{mid}(e) \mid H' \in \mathcal{C}(H)\} = \mathcal{A} \;\wedge \\ & \quad \{(v, \mathbf{deg}_H(v)) \mid v \in \cup_{A \in \mathcal{A}} A\} = \psi \\ & \text{else if } (\mathcal{A} = \emptyset) \text{ then} \\ & \quad |\mathcal{C}(H)| \leq 1 \;\wedge\; V(H) \cap \mathbf{mid}(e) = \emptyset \;\}\}\end{aligned}$$

Clearly, $\mathbf{opt}_{e_r}(\emptyset, \varnothing) = \mathbf{medbcs}_d(G)$. The idea is the following:

- If $\mathcal{A} \neq \emptyset$, we look for the best solution H in the graph G_e such that its restriction to $\mathbf{mid}(e)$ induces the connected components given by $\mathcal{A}$ and obeys the degrees given by ψ.
- Otherwise, if $\mathcal{A} = \emptyset$, we look for the best solution H in G_e not intersecting $\mathbf{mid}(e)$. Since $\mathbf{mid}(e)$ is a separator of G, it is clear that in this case the solution H must be a connected subgraph of G_e disjoint from $\mathbf{mid}(e)$.

Let us now see how these values of $\mathbf{opt}_e(\mathcal{A}, \psi)$ can be explicitly computed using dynamic programming over a branch decomposition of G.

Let e, e_1, e_2 be three edges of T that are incident to the same vertex and such that e is closer to the root of T than the other two (see the upper part of Figure 2). To perform the *join/forget* operations in the middle set $\mathbf{mid}(e)$, we distinguish two cases according to the packing $\mathcal{A}$ of $\mathbf{mid}(e)$:

(1) In the case $\mathcal{A} \neq \emptyset$, the value of $\mathbf{opt}_e(\mathcal{A}, \psi)$ is given by

$$\begin{aligned}\mathbf{opt}_e(\mathcal{A}, \psi) \;=\; \max\{\{0\} \cup \{l : \exists & (\mathcal{A}_i, \psi_i) \in \mathscr{P}_{e_i}, i = 1, 2, \text{ such that} \\ & \cup \mathcal{A}_1 \cap (\mathbf{mid}(e_1) \cap \mathbf{mid}(e_2)) = \cup \mathcal{A}_2 \cap (\mathbf{mid}(e_1) \cap \mathbf{mid}(e_2)) \wedge \\ & (\mathcal{A}_1, \psi_1) \oplus (\mathcal{A}_2, \psi_2) \text{ is a } d\text{-weighted} \\ & \quad \text{packing of } \mathbf{mid}(e_1) \cup \mathbf{mid}(e_2) \;\wedge \\ & (\mathcal{A}, \psi) = ((\mathcal{A}_1, \psi_1) \oplus (\mathcal{A}_2, \psi_2))|_{\mathbf{mid}(e)} \;\wedge \\ & \text{if } (\mathcal{A}_1 = \emptyset) \text{ then } l = \mathbf{opt}_{e_2}(\mathcal{A}_2, \psi_2) \\ & \text{if } (\mathcal{A}_2 = \emptyset) \text{ then } l = \mathbf{opt}_{e_1}(\mathcal{A}_1, \psi_1) \\ & \text{else } l = \mathbf{opt}_{e_1}(\mathcal{A}_1, \psi_1) + \mathbf{opt}_{e_2}(\mathcal{A}_2, \psi_2) \;\}\}\end{aligned}$$

(2) In the case $\mathcal{A} = \emptyset$, the value of $\mathbf{opt}_e(\emptyset, \psi)$ is given by

$$
\begin{aligned}
\mathbf{opt}_e(\emptyset,\psi) \;=\; \max\{\{0\} \cup \{l : \exists (\mathcal{A}_i,\psi_i) \in \mathscr{P}_{e_i}, i=1,2, \text{ such that}\\
&\cup \mathcal{A}_1 \cap (\mathbf{mid}(e_1)\cap\mathbf{mid}(e_2)) = \cup\mathcal{A}_2 \cap (\mathbf{mid}(e_1)\cap\mathbf{mid}(e_2)) \wedge \\
&(\mathcal{A}_1,\psi_1)\oplus(\mathcal{A}_2,\psi_2) \text{ is a } d\text{-weighted}\\
&\quad \text{packing of } \mathbf{mid}(e_1)\cup\mathbf{mid}(e_2) \ \wedge\\
&(\emptyset,\psi) = ((\mathcal{A}_1,\psi_1)\oplus(\mathcal{A}_2,\psi_2))|_{\mathbf{mid}(e)} \ \wedge\\
&\text{if } (\mathcal{A}_1=\emptyset \ \wedge \ \mathcal{A}_2=\emptyset) \text{ then}\\
&\quad l = \max\{\mathbf{opt}_{e_1}(\mathcal{A}_1,\psi_1), \mathbf{opt}_{e_2}(\mathcal{A}_2,\psi_2)\}\\
&\text{if } (\mathcal{A}_1\neq\emptyset \ \wedge \ \mathcal{A}_2=\emptyset) \text{ then}\\
&\quad l = \max\{\mathbf{opt}_{e_2}(\mathcal{A}_2,\psi_2), \{\mathbf{opt}_{e_1}(\mathcal{A}_1,\psi_1)|_X : \ X\in\mathcal{A}_1\}\}\\
&\text{if } (\mathcal{A}_1=\emptyset \ \wedge \ \mathcal{A}_2\neq\emptyset) \text{ then}\\
&\quad l = \max\{\mathbf{opt}_{e_1}(\mathcal{A}_1,\psi_1), \{\mathbf{opt}_{e_2}(\mathcal{A}_2,\psi_2)|_X : \ X\in\mathcal{A}_2\}\}\\
&\text{if } (\mathcal{A}_1\neq\emptyset \ \wedge \ \mathcal{A}_2\neq\emptyset) \text{ then}\\
&\quad l = \max\{\mathbf{opt}_{e_1}(X,\psi_1)|_{\mathbf{mid}(e_1)} + \mathbf{opt}_{e_2}(X,\psi_2)|_{\mathbf{mid}(e_2)} :\\
&\qquad\qquad X \in \mathcal{C}(\mathbf{mid}(e_1)\cup\mathbf{mid}(e_2), \mathcal{A}_1\cup\mathcal{A}_2)\}\ \}\}
\end{aligned}
$$

These ideas are schematically illustrated in Figure 2. So far, we have shown how to compute $\mathbf{opt}_e(\mathcal{A}, \psi)$ for e being an internal edge of T. Finally, suppose that $e_{\text{leaf}} = \{x, y\} \in E(T)$ is an edge such that either x or y is a leaf of T. Let $\{v_1, v_2\} \in E(G)$ be the image under μ of the endpoint of e which is a leaf of T. Then

$$\mathbf{opt}_{e_{\text{leaf}}}(\mathcal{A},\psi) = \begin{cases} 1 & \text{, if } \ (\mathcal{A} = \{\{v_1,v_2\}\} \quad \wedge \quad \psi = \{(v_1,1),(v_2,1)\}) \\ 0 & \text{, otherwise} \end{cases}$$

Running time. The size of the tables of the dynamic programming over the branch decomposition of the input graph, namely $|\mathscr{P}_e|$, determines the running time of our algorithms. The number of ways a set of h elements can be partitioned into nonempty subsets is well-known as the h-th *Bell number* [8] and is denoted by B_h. We can express $|\mathscr{P}_e|$ in terms of the Bell numbers:

$$|\mathscr{P}_e| \;=\; (d+1)^h \cdot \sum_{i=0}^{h} \binom{h}{i} B_{h-i} \;\leq\; (d+1)^h \cdot 2^{2h\cdot\log h}, \qquad (1)$$

where the last inequality is an easy exercise using that $B_h \leq \frac{e^h - 1}{(\log h)^h} h!$ [8]. At each edge e of the branch decomposition, to compute all the values $\mathbf{opt}_e(\mathcal{A}, \psi)$ we test all the possibilities of combining d-weighted packings of the two middle sets $\mathbf{mid}(e_1)$ and $\mathbf{mid}(e_2)$. The operations $(\mathcal{A}_1, \psi_1)\oplus(\mathcal{A}_2, \psi_2)$ and $(\mathcal{A}, \psi)|_{A'}$ take $\mathcal{O}(|\mathbf{mid}(e)|)$ time. Let $m = |E(G)|$. Hence, by Equation (1), given a branch decomposition of a general graph G of width at most h, the value of $\mathbf{medbcs}_d(G)$ can be computed in $(d + 1)^{2h} \cdot 2^{4h\cdot\log h} \cdot h \cdot m$ steps.

5 Speed-up for Planar Graphs using Catalan Structures

In this section we will see that when the input is restricted to planar graphs the term $2^{\mathcal{O}(h\cdot\log h)}$ in Equation (1) can be reduced to $2^{\mathcal{O}(h)}$. Our analysis is inspired from [6].

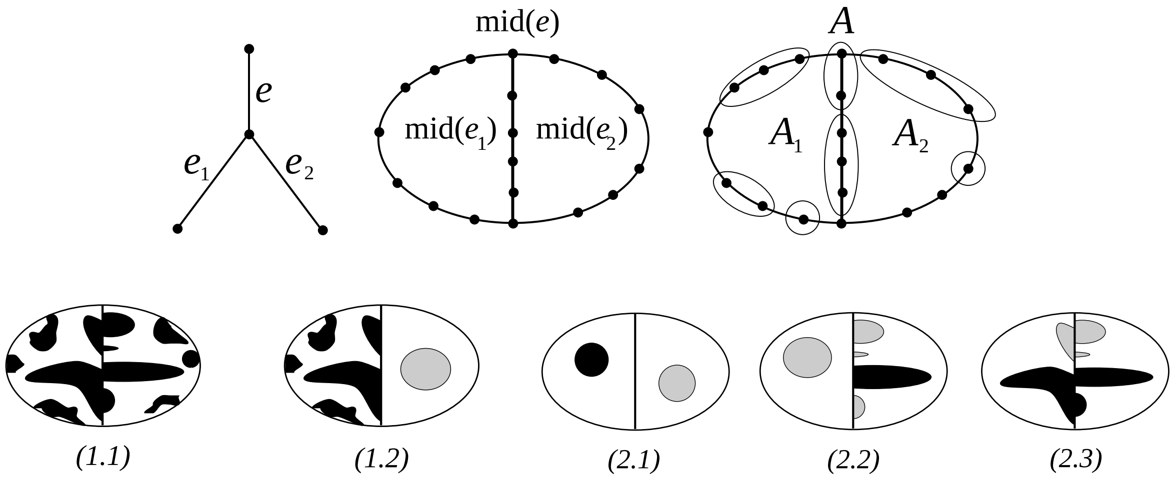

Figure 2: *Join/forget* operations in the dynamic programming over a branch decomposition. On the upper part, the vertices around the external oval belong to $\mathbf{mid}(e)$, and separate G_e (inside) from $G \setminus G_e$ (outside). The vertices around the smaller left (resp. right) oval belong to $\mathbf{mid}(e_1)$ (resp. $\mathbf{mid}(e_2)$). In the rightmost figure, these vertices are grouped according to $\mathcal{A}$. On the lower part, the dark regions represent an optimal subgraph in each case, while the grey regions represent the partial solutions discarded by the algorithm. Case *(1)*: $\mathcal{A} \neq \emptyset$; *(1.1)* $\mathcal{A}_1 \neq \emptyset, \mathcal{A}_2 \neq \emptyset$; *(1.2)* $\mathcal{A}_1 \neq \emptyset, \mathcal{A}_2 = \emptyset$. Case *(2)*: $\mathcal{A} = \emptyset$; *(2.1)* $\mathcal{A}_1 = \emptyset, \mathcal{A}_2 = \emptyset$; *(2.2)* $\mathcal{A}_1 = \emptyset, \mathcal{A}_2 \neq \emptyset$; *(2.3)* $\mathcal{A}_1 \neq \emptyset, \mathcal{A}_2 \neq \emptyset$.

Let G be a planar graph embedded on a sphere $\mathbb{S}$. An *O-arc* is a subset of $\mathbb{S}$ homeomorphic to a circle. An O-arc in S is called a *noose* of the embedding of G if it meets G only in vertices. A *sphere cut decomposition* or *sc-decomposition* (T, μ, π) of G is a branch decomposition of G with the following property: for every edge e of T, there exists a noose O_e meeting every face at most once and 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 $\mathbf{mid}(e)$ and its length is $|\mathbf{mid}(e)|$. A *clockwise traversal* of O_e in the embedding of G defines the cyclic ordering π of $\mathbf{mid}(e)$. (More details can be found in [20].) We always assume that the vertices of every middle set $\mathbf{mid}(e) = V(G_1) \cap V(G_2)$ are enumerated according to π.

Theorem 2 (Seymour and Thomas [20]) *Let G be a planar graph of branchwidth at most h without vertices of degree one embedded on a sphere. Then there exists an sc-decomposition of G of width at most h.*

In addition, such an sc-decomposition can be constructed in time $\mathcal{O}(n^3)$ [14]. See [15] for recent improvements. The size of the tables of the dynamic programming algorithm is given by the number of ways a solution of k-MDBCS$_d$ in G_e can intersect $\mathbf{mid}(e)$. Let (T, μ, π) be a sphere cut decomposition of width at most h, and we can assume $h \leq \mathbf{bw}(G)$ by Theorem 2. Then the vertices of $\mathbf{mid}(e)$ are situated around a noose. A *non-crossing partition (ncp)* of a cyclically ordered set

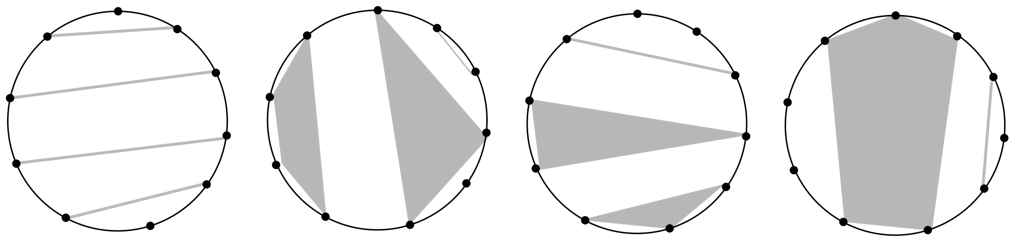

Figure 3: Catalan structures in the middle set of a sphere cut decomposition.

$S = \{1, \ldots, h\}$ is a partition $\{P_1, \ldots, P_m\}$ of S such that there are no numbers $a < b < c < d$ where $a, c \in P_i$, and $b, d \in P_j$ with $i \neq j$.

When we restrict the input graph G to be planar, then the subgraph given by the intersection of a partial solution of k-MDBCS$_d$ in G_e with $\mathbf{mid}(e)$ is also planar. We can think of each connected component of this subgraph as a virtual hyperedge among vertices in $\mathbf{mid}(e)$. The reduction from $2^{\mathcal{O}(h \cdot \log h)}$ to $2^{\mathcal{O}(h)}$ is based on an estimate of the number of ways we can draw hyperedges inside a cycle such that they touch the cycle on its vertices and they do not share common internal points in the plain (they do not intersect), as it is illustrated in Figure 3.

The number of such configurations is closely related to the number of *non-crossing partitions* over h vertices, which is equal to the h-th *Catalan number* $\mathrm{CN}(h) = \frac{1}{h+1}\binom{2h}{h} \sim \frac{4^h}{\sqrt{\pi} h^{3/2}} \leq 4^h$ [16].

Indeed, in the same spirit of Equation (1) we can write

$$
\begin{aligned}
|\mathscr{P}_e| &= (d+1)^h \cdot \sum_{i=0}^{h} \binom{h}{i} \mathrm{CN}(h-i) \;\leq\; (d+1)^h \cdot \sum_{i=0}^{h} \binom{h}{i} 4^{h-i} \\
&= (d+1)^h 4^h \cdot \sum_{i=0}^{h} \binom{h}{i} \left(\frac{1}{4}\right)^i \;=\; (d+1)^h 4^h \cdot \left(1 + \frac{1}{4}\right)^h = (d+1)^h \cdot 5^h.
\end{aligned}
$$

Since G is planar, $|E(G)| = \mathcal{O}(|V(G)|)$, hence so is the number of middle sets in any branch decomposition of G. Therefore,

Proposition 1 *For every planar graph G and given a sphere cut decomposition (T, μ, π) of G of width $\leq h$, the value of $\mathbf{medbcs}_d(G)$ can be computed in $\mathcal{O}\left((d+1)^{2h} \cdot 5^{2h} \cdot h \cdot n\right)$ steps.*

Let δ be the constant defined in Lemma 3. Summarizing,

Theorem 3 k-Planar Maximum d-Degree-Bounded Connected Subgraph *is solvable in time $\mathcal{O}\left(2^{\log(5(d+1)) 8\sqrt{k}/\delta} \sqrt{k} \cdot n + n^3\right)$ for any $d \geq 2$.*

Proof: First, using Theorem 2, we construct in time $\mathcal{O}(n^3)$ an optimal sphere cut decomposition of G of width $\mathbf{bw}(G)$. We distinguish two cases: If $\mathbf{bw}(G) > 4/\delta \cdot \sqrt{k}$, then by Lemma 3 the answer to the parameterized problem is automatically YES. Otherwise, $\mathbf{bw}(G) \leq 4/\delta \cdot \sqrt{k}$ and the value of $\mathbf{medbcs}_d(G)$ can be computed by Proposition 1 in time $\mathcal{O}\left((d+1)^{8\sqrt{k}/\delta} \cdot 5^{8\sqrt{k}/\delta} \cdot 4/\delta\sqrt{k} \cdot n\right) =$

$\mathcal{O}\left(2^{\log(5(d+1))8\sqrt{k}/\delta}\sqrt{k}\cdot n\right)$. ∎

It is worth mentioning that the algorithms of [6] for the LONGEST PATH problem can be easily adapted to deal with the case $d = 2$ of MAXIMUM d-DEGREE-BOUNDED CONNECTED SUBGRAPH. This yields an algorithm with running time $\mathcal{O}\left(2^{13.6\sqrt{k}}\sqrt{k}\cdot n + n^3\right)$, which improves over the running time of Theorem 3 for the specific case $d = 2$.

6 Extensions

Appropriate modifications of the dynamic programming algorithm of Section 4 allow us to obtain also subexponential parameterized algorithms for the variant of the problem in which the aim is to maximise the number of vertices of the subgraph H, as well as for the variant in which the output subgraph is required to be induced (for both the edge and vertex maximisation versions). Another variant is when the list of prescribed degrees of the vertices belongs to a subset of $\mathbb{Z}_q$ for a fixed integer q. Finally, we discuss how to transform these parameterized algorithms into subexponential *exact* algorithms on planar graphs.

6.1 Maximising the number of vertices

In this section we focus on the following family of problem for $d \geq 2$:

> VERTEX MAXIMUM d-DEGREE-BOUNDED CONNECTED SUBGRAPH (VMDBCS$_d$)
> **Input:** A graph G and a non-negative integer k.
> **Question:** Does G contain a connected subgraph H with $\Delta(H) \leq d$ and $|V(H)| \geq k$?

In order to obtain subexponential parameterized algorithms for VMDBCS$_d$ on planar graphs, let us see how the techniques presented in the preceding sections must be modified. The corresponding parameter is

$$\mathbf{mvdbcs}_d(G) = \max\{|V(H)| \mid H \subseteq G \ \wedge \ H \text{ is connected } \wedge \ \Delta(H) \leq d\}.$$

First, it is easy to check that Lemmas 2 and 3 hold for the parameter $\mathbf{mvdbcs}_d(G)$ with $\delta = 1$ for any $d \geq 2$. Secondly, the dynamic programming approach of Section 4 remains the same, except for the following modifications.

When computing a partial solution $\mathbf{opt}_e(\mathcal{A}, \psi)$ in G_e from the partial solutions in G_{e_1} and G_{e_2}, we have to be careful in order to avoid counting twice the vertices that belong to both $\mathbf{mid}(e_1)$ and $\mathbf{mid}(e_2)$. More precisely,

- in the case $\mathcal{A} \neq \emptyset$, $\mathcal{A}_1 \neq \emptyset$, and $\mathcal{A}_2 \neq \emptyset$ we have that

  $$l = \mathbf{opt}_{e_1}(\mathcal{A}_1, \psi_1) + \mathbf{opt}_{e_2}(\mathcal{A}_2, \psi_2) - |V(\mathcal{G}[\mathcal{A}_1]) \cap V(\mathcal{G}[\mathcal{A}_2])|,$$

  where $\mathcal{G}[\mathcal{A}_i]$, $i = 1, 2$, denotes the hypergraph induced by the hyperedges in $\mathcal{A}_i$; and

- in the case $\mathcal{A}=\emptyset$, $\mathcal{A}_1 \neq \emptyset$, and $\mathcal{A}_2 \neq \emptyset$ we have that

$$\begin{aligned} l &= \max\{\mathbf{opt}_{e_1}(X,\psi_1)|_{\mathbf{mid}(e_1)} + \mathbf{opt}_{e_2}(X,\psi_2)|_{\mathbf{mid}(e_2)} - \\ & \quad |V(\mathcal{G}[X]) \cap \mathbf{mid}(e_1) \cap \mathbf{mid}(e_2)| \ : \\ & \quad X \in \mathcal{C}(\mathbf{mid}(e_1) \cup \mathbf{mid}(e_2), \mathcal{A}_1 \cup \mathcal{A}_2)\} \ \}. \end{aligned}$$

Also, if $e_{\text{leaf}} = \{x,y\} \in E(T)$ is an edge such that x is a leaf of T, and $\{v_1,v_2\} \in E(G)$ is the image of x under μ, then

$$\mathbf{opt}_{e_{\text{leaf}}}(\mathcal{A},\psi) = \begin{cases} 2 & \text{, if } (\mathcal{A}=\{\{v_1,v_2\}\} \quad \wedge \quad \psi=\{(v_1,1),(v_2,1)\}) \\ & \quad \vee\, (\mathcal{A}=\{\{v_1\},\{v_2\}\} \quad \wedge \quad \psi=\{(v_1,0),(v_2,0)\}) \\ 1 & \text{, if } (\mathcal{A}=\{\{v_1\}\} \quad \wedge \quad \psi=\{(v_1,0),(v_2,0)\}) \\ & \quad \vee\, (\mathcal{A}=\{\{v_2\}\} \quad \wedge \quad \psi=\{(v_1,0),(v_2,0)\}) \\ 0 & \text{, otherwise} \end{cases}$$

Finally, the speed-up described in Section 5 can be directly appliedt o VMDBCS$_d$, since Catalan structures also appear in the middle sets of a sc-decomposition of the planar input graph. Summarizing,

Theorem 4 k-PLANAR VERTEX MAXIMUM d-DEGREE-BOUNDED CONNECTED SUBGRAPH *is solvable in time* $\mathcal{O}\left(2^{\log(5(d+1))8\sqrt{k}}\sqrt{k}\cdot n + n^3\right)$ *for any* $d \geq 2$.

6.2 Looking for an induced subgraph

It is also natural to ask, instead of for a subgraph H of the input graph G, for an *induced* subgraph H. In this section we focus on the edge-maximisation version of the problem, the modifications for the node-maximisation version being analogous to those described in Section 6.1. We denote the problem by MAXIMUM d-DEGREE-BOUNDED CONNECTED INDUCED SUBGRAPH (MDBCIS$_d$).

In contrast to the dynamic programming presented in Section 4, now we need only to consider those packings $\mathcal{A}$ of $\mathbf{mid}(e)$ that "respect" the fact that the solution subgraph must be induced. Namely, if two adjacent vertices v_1, v_2 belong to a partial solution, then the edge $\{v_1,v_2\}$ must also belong to the solution. This property can be incorporated in the algorithm of Section 4 by just imposing it in the leaves of the branch decomposition. Indeed, if in a leaf corresponding to an edge $\{v_1,v_2\} \in E(G)$ we forbid the packing $\mathcal{A}=\{\{v_1\},\{v_2\}\}$, then all the partial solutions will be induced subgraphs. Therefore, the values in the leaves must be updated to

$$\mathbf{opt}_{e_{\text{leaf}}}(\mathcal{A},\psi) = \begin{cases} 1 & \text{, if } (\mathcal{A}=\{\{v_1,v_2\}\} \quad \wedge \quad \psi=\{(v_1,1),(v_2,1)\}) \\ 0 & \text{, if } (\mathcal{A}=\{\{v_1\}\} \quad \wedge \quad \psi=\{(v_1,0),(v_2,0)\}) \\ & \quad \vee\, (\mathcal{A}=\{\{v_2\}\} \quad \wedge \quad \psi=\{(v_1,0),(v_2,0)\}) \\ & \quad \vee\, (\mathcal{A}=\emptyset \quad \wedge \quad \psi=\{(v_1,0),(v_2,0)\}) \\ \texttt{unfeasible} & \text{, otherwise} \end{cases}$$

When combining partial solutions from two middle sets $\mathbf{mid}(e_1)$ and $\mathbf{mid}(e_2)$, we must take only into account those pairs which "agree", that is, those which coincide in $\mathbf{mid}(e_1) \cap \mathbf{mid}(e_2)$. The rest of the algorithm of Section 4 remains the same. Finally, the constant δ of Theorem 3 must be replaced with $\delta' = \delta/\sqrt{2}$ when $d \in \{2,3\}$, due to the fact that the optimal subgraphs of MDBCIS_d on the square grid (see Lemma 2) must be induced. Summarizing,

Theorem 5 *k-PLANAR MAXIMUM d-DEGREE-BOUNDED CONNECTED INDUCED SUBGRAPH is solvable in time $\mathcal{O}\left(2^{\log(5(d+1))8\sqrt{k}/\delta'} \cdot \sqrt{k} \cdot n + n^3\right)$ for any $d \geq 2$.*

6.3 More general constraints on the degree

All the variants of the problem considered so far have in common that the degree of any vertex belonging to the output subgraph must lie in the interval $[0,d]$. It makes sense to consider a more general version in which the interval of allowed degrees depends on each vertex. Namely, for each vertex $v \in V(G)$ we are given an interval $I_v = [h_v, r_v]$ and we look for a maximum connected subgraph H in which the degree of each vertex v lies in I_v. (If $0 \in I_v$ then vertex v may not belong to $V(H)$.) When the output subgraph is not required to be connected, some variants of the problem are in P and some others become NP-hard [17]. In general, we cannot guarantee that the parameters associated with this general problem are minor closed, hence the approach used with MDBCS_d does not carry over. Nevertheless, we can obtain an algorithm to solve it similar to the one of Proposition 1, replacing the term $(d+1)^{2h}$ with $(\max_{v \in V(G)} r_v + 1)^{2h}$. The ideas behind the dynamic programming are essentially the same.

Another variant is obtained when forcing the allowed degrees to belong to a subset of $\mathbb{Z}_q$ for some fixed integer q. In this case it is not difficult to see that the term $(d+1)^{2h}$ can be replaced with q^{2h}. For instance, the case where all the degrees are required to be 0 (mod 2) corresponds to the MAXIMUM EULERIAN SUBGRAPH problem. This approach, given a planar graph with a sphere cut decomposition of width at most h, yields an algorithm to solve MAXIMUM EULERIAN SUBGRAPH in time $\mathcal{O}\left(2^{2h} \cdot 5^{2h} \cdot h \cdot n\right)$.

6.4 Exact algorithms

The subexponential parameterized algorithms we have presented on planar graphs can be naturally transformed to subexponential *exact* algorithms by using that for any planar graph G, $\mathbf{bw}(G) \leq \sqrt{4.5 \cdot |V(G)|}$ [10].

Indeed, given a planar graph G and a sphere cut decomposition of width at most $\sqrt{4.5 \cdot |V(G)|}$, we can compute an optimal solution of MDBCS_d in G in $\mathcal{O}\left((d+1)^{4.24\sqrt{n}} \cdot 5^{4.24\sqrt{n}} \cdot n^{3/2}\right)$ steps (by Proposition 1). The same argument applies to all the variants of the problem discussed above.

In addition, we can derive a subexponential exact algorithm for the following problem on planar graphs: MINIMUM DEGREE SPANNING TREE (MDST). In the MDST problem, given an undirected unweighted graph G, the objective is to find a spanning tree of G which minimizes the maximum degree over all the spanning trees of G. This problem has been widely studied in the literature (cf. for instance [12]), and we are unaware of the existence of subexponential exact algorithms on planar graphs. Our algorithm works as follows: given a planar graph G, we find an optimal solution H_d of VMDBCS_d in G for $d = 2, \ldots, n-1$. Let d^* be the first value of d for which $|V(H_d)| = n$. Then an optimal solution of MDST in G is given by any spanning tree of H_{d^*}.

A graph is *supereulerian* if it has a spanning Eulerian subgraph [2]. Combining the ideas of the algorithm above with the ideas of Section 6.3 yields a subexponential exact algorithm to decide whether a planar graph is supereulerian or not.

7 Conclusions

In this article we obtained a $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ algorithm for k-MDBCS$_d$ and related problems on planar graphs, following the approach of [3–6]. Several interesting problems remain open. First, it seems natural to try to improve the worst-case running time of our algorithms. Much more challenging is to find subexponential parameterized algorithms for the edge- or node-weighted versions of the problem. Actually, the weighted versions of our parameters remain minor closed (by an easy modification of Lemma 2), however the fundamental difference is that the combinatorial bound of Lemma 3 does not hold anymore. On the other hand, the natural extension of this article would be to conceive subexponential parameterized algorithms for k-MDBCS$_d$ on other sparse graph classes, like graphs of bounded genus and, more generally, minor-free families of graphs.

Finally, note that the MDBCS$_d$ problem is equivalent to finding a maximum connected subgraph not containing the star $K_{1,d+1}$ as a topological minor. Many classical NP-hard problems can be expressed as finding a maximum subgraph excluding a fixed graph H as a minor (or induced minor, or subgraph, or induced subgraph, or topological minor), hence conceiving a general framework to design subexponential parameterized algorithms for this class of problems would be a celebrated result.

Acknowledgement. We would like to thank the anonymous referees for helpful comments that improved the presentation of the article.

References

[1] O. Amini, D. Peleg, S. Pérennes, I. Sau, and S. Saurabh. Degree-constrained subgraph problems: Hardness and approximation. In *Proc. of the 6th Workshop on Approximation and On-line Algorithms (WAOA)*, volume 5426 of *LNCS*, pages 29–42, 2008.

[2] P. A. Catlin. Supereulerian graphs: a survey. *Journal of Graph Theory*, 16(2):177–196, 1992.

[3] 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. *Journal of the ACM*, 52(6):866–893, 2005.

[4] F. Dorn, F. V. Fomin, and D. M. Thilikos. Subexponential parameterized algorithms. In *Proc. of the 34th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4596 of *LNCS*, pages 15–27, 2007.

[5] F. Dorn, F. V. Fomin, and D. M. Thilikos. Catalan structures and dynamic programming in H-minor-free graphs. In *Proc. of the 19th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 631–640, 2008.

[6] F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In *Proc. of the 13th Annual European Symposium on Algorithms (ESA)*, volume 3669 of *LNCS*, pages 95–106, 2005.

[7] F. Dorn and J. A. Telle. Semi-nice tree-decompositions: The best of branchwidth, treewidth and pathwidth with one algorithm. *Discrete Applied Mathematics*, 157(12):2737–2746, 2009.

[8] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2008.

[9] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer Verlag, 2006.

[10] F. V. Fomin and D. M. Thilikos. New upper bounds on the decomposability of planar graphs. *Journal of Graph Theory*, 51(1):53–81, 2005.

[11] F. V. Fomin and D. M. Thilikos. Dominating Sets in Planar Graphs: Branch-Width and Exponential Speed-Up. *SIAM Journal on Computing*, 36(2):281–309, 2006.

[12] M. Fürer and B. Raghavachari. Approximating the minimum-degree spanning tree to within one from the optimal degree. In *Proc. of the 3rd annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 317–324, 1992.

[13] M. Garey and D. Johnson. *Computers and Intractability.* W.H. Freeman, 1979.

[14] Q.-P. Gu and H. Tamaki. Optimal branch-decomposition of planar graphs in $O(n^3)$ time. *ACM Transactions on Algorithms*, 4(3), 2008.

[15] Q.-P. Gu and H. Tamaki. Constant-factor approximations of branch-decomposition and largest grid minor of planar graphs in $O(n^{1+\varepsilon})$ time. In *Proc. of the 20th International Symposium Algorithms and Computation (ISAAC)*, volume 5878 of *LNCS*, pages 984–993, 2009.

[16] G. Kreweras. Sur les partitions non croisées d'un cercle. *Discrete Mathematics*, 1:333–350, 1972.

[17] L. Lovász and M. Plummer. *Matching Theory.* Annals of Discr. Math. 29, 1986.

[18] N. Robertson and P. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Comb. Theory, Series B*, 52(2):153–190, 1991.

[19] N. Robertson, P. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Comb. Theory Ser. B*, 62(2):323–348, 1994.

[20] P. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.