Pathlength of Outerplanar Graphs

Thomas Dissaux — Université Côte d'Azur, Inria, CNRS, I3S, France
Nicolas Nisse — Université Côte d'Azur, Inria, CNRS, I3S, France

Abstract

A *path-decomposition* of a graph $G = (V, E)$ is a sequence of subsets of V, called *bags*, that satisfy some connectivity properties. The *length* of a path-decomposition of a graph G is the greatest distance between two vertices that belong to a same bag and the *pathlength*, denoted by $p\ell(G)$, of G is the smallest length of its path-decompositions. This parameter has been studied for its algorithmic applications for several classical metric problems. Deciding if the pathlength of a graph G is at most 2 is NP-complete, however no result about planar graphs is known. In this work, we first show that $p\ell(C_n) = \lfloor \frac{n}{2} \rfloor$ for any n-cycle C_n and that the pathlength can be computed in linear time in trees. Our main result is a (+1)-approximation algorithm for computing the pathlength of 2-connected outerplanar graphs, based on a characterization of almost optimal (of length at most $p\ell(G) + 1$) path-decompositions of such graphs.

1 Introduction

Path-decompositions of graphs are well studied since their introduction in the Graph Minor Theory of Robertson and Seymour, and for their various algorithmic applications. A *path-decomposition* of a graph $G = (V, E)$ is a sequence $(X_1, \ldots, X_p)$ of subsets of V (called *bags*) such that (1) $\bigcup_{i \le p} X_i = V$, (2) for every edge $\{u, v\} \in E$, there exists $1 \le i \le p$ such that $u, v \in X_i$, and (3) for all $1 \le i \le z \le j \le p$, $X_i \cap X_j \subseteq X_z$. These properties imply another fundamental (in proofs) property, which is that for all $1 \le i < p$, $S = X_i \cap X_{i+1}$ *separates* $\bigcup_{j \le i} X_j \setminus S$ and $\bigcup_{j \ge i} X_j \setminus S$.

One of the most famous measure about path-decompositions is the *width* (corresponding to the maximum size of the bags minus one). The *pathwidth* of a graph G is the minimum width of its path-decompositions. Typically, the famous Courcelles's theorem implies that various NP-hard (in general graphs) problems can be solved in polynomial time in graphs with bounded pathwidth.

We focus on another measure (less studied) of path-decompositions which has also various algorithmic applications. For all $u, v \in V$, let $d_G(u, v)$ (or $d(u, v)$ if there is no ambiguity) be the distance between u and v in G. The *length* $\ell(D)$ of a path-decomposition $D = (X_1, \ldots X_p)$ of G is the maximum *diameter* of its bags, i.e., $\ell(D) = \max_{i \le p} \max_{u,v \in X_i} d_G(u, v)$. The *pathlength* $p\ell(G)$ of a graph G is the minimum length of its path-decompositions [6].

Concerning the applications, the line-distortion problem can be approximated in graphs of bounded pathlength [7], which has applications in computer vision, computational chemistry and biology, distributed protocols...(see references in [8]). Moreover, the pathlength being an upper bound of the treelength, several problems like the traveling salesman problem or the metric dimension can be solved (or approximated) efficiently in graphs of bounded pathlength [9, 1, 4].

For any graph G, deciding if $p\ell(G) \le 2$ is NP-complete and there is no c-approximation of $p\ell(G)$ for all $c < 3/2$ (if $P \ne NP$) [8]. The best known approximation of $p\ell(G)$ has a factor 2 [7] but the complexity in the case of planar graphs is unknown. Here, we initiate the study of planar graphs by first considering *outerplanar* graphs (graphs that can be embedded in a plane without crossing edges and such that every vertex is in the outer face). Note that this class of graphs has already been studied in the cases of treelength [6] and of pathwidth. In the latter case, the best exact algorithm has time complexity at least $O(|V|^{11})$ and there exists a 2-approximation [2, 3].

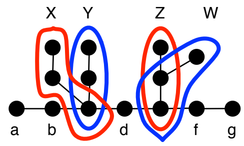

Figure 1: Example of a tree T (with $p\ell(T) = 2$) where one of the optimal path-decompositions described in Theorem 1 is $(\{ab\}, \{bc\}, X, Y, \{cd\}, \{de\}, Z, W, \{ef\}, \{fg\})$.

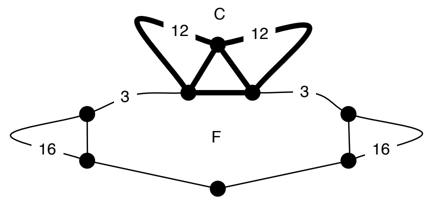

Figure 2: Example of a graph G with no optimal path-decomposition ($p\ell(G) = 10$) satisfying Property 2 for the component C (in bold). A line labeled x is a path with x edges.

Contributions. We first give a linear-time algorithm that computes the pathlength of trees, and prove that $p\ell(C_n) = \lfloor \frac{n}{2} \rfloor$ for all cycle C_n with n vertices. Our main result is an algorithm that computes, in time $O(n^3 \cdot p\ell(G)^2)$, a path-decomposition of length at most $p\ell(G) + 1$ of any 2-connected outerplanar graph G. Note that proofs have been omitted but can be found here [5].

Preliminaries. A *diameter* of a graph $G = (V, E)$ is any shortest path with maximum length ($\max_{x,y \in V} d(x, y)$) in G. The distance $d(u, S)$ from $u \in V$ to $S \subseteq V$ equals $\min_{v \in S} d(u, v)$. From now on, "decomposition" will always refer to path-decomposition. The following property is widely used below. A subgraph H of a graph G is *isometric* if, for all $u, v \in V(H)$, $d_H(u, v) = d_G(u, v)$ (distances from G are preserved in H). For any isometric subgraph H of G, $p\ell(H) \leq p\ell(G)$ [6].

2 Pathlength of trees, cycles and outerplanar graphs

We begin by investigating the pathlength of trees. This result relatively simple will give us an intuition on how the pathlength of outerplanar graphs can be computed.

Theorem 1. *Let $T = (V, E)$ be any tree that is not a path. Let P be a diameter of T and let $k = \max_{v \in V} d(v, V(P)) > 0$. Then, $p\ell(T) = k$ can be computed in linear time.*

Sketch of proof : Since T is not a path, T contains a *k-spider* graph (which is obtained from $K_{1,3}$ by subdividing each edge $k - 1$ times) as isometric subgraph and so $p\ell(G) \geq k$ [6]. For the upper bound, we define a decomposition D of T of length k. Intuitively, this decomposition "follows" P by "adding" the "branches" sequentially in the order they are met. Formally, let $P = (v_0, \ldots, v_r)$, $e_i = \{v_{i-1}, v_i\}$ and T_i is the component of $T \setminus \{e_i, e_{i+1}\}$ containing v_i ($0 \leq i < r$). The first bag of D is $\{v_0, v_1\}$, then sequentially (from $i = 1$ to $r - 1$), we "add" the bags containing respectively the path from v_i to a leaf of T_i (for all leafs of T_i, ordered by an arbitrary DFS of T_i starting at v_i), followed by the bag containing $\{v_i, v_{i+1}\}$ (see an illustration in Figure 1). Clearly, P and a such decomposition (and $p\ell(G)$) can be computed in linear time. $\diamond$

We now look at 2-connected outerplanar graphs. Let us begin with the simplest of them, the cycles.

Theorem 2. *[5] Let $C_n = (V, E)$ be a cycle with $n \geq 3$ vertices. Then, $p\ell(C_n) = \lfloor \frac{n}{2} \rfloor$.*

Let us consider a 2-connected outerplanar graph $G = (V, E)$. The *weak dual* G^* of G is the graph whose vertices are the internal (bounded) faces of G and two vertices of G^* are adjacent if their corresponding faces share an edge in G. Note that, by 2-connectivity, G has a unique outerplanar embedding and G^* is a tree (see Fig. 3). By following the idea of the proof of Theorem 1, for

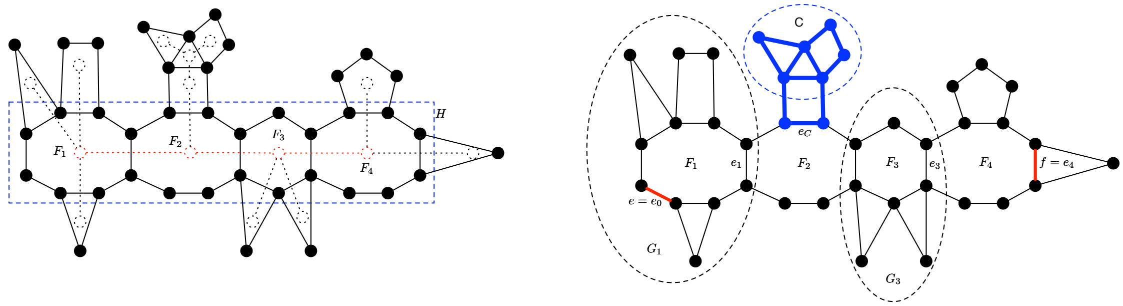

Figure 3: Notations of Theorem 3 on an example. On the left, a 2-connected outerplanar graph G (in solid lines), its weak dual G^* (in dotted lines) with the path P^* (in red) associated to the edges e and f and the corresponding "path of faces" H (in blue). On the right, the subgraphs G_1 and G_3, and the subgraph $C \cup e_C$ (in blue) associated to the component C of $G \backslash H$ are illustrated.

any path P^* of G^*, a decomposition (not necessarily optimal) of the tree G^* can be defined by "following" P^* and by "adding" the "branches" (of $G^* \backslash P^*$) in the order they are met.

We "transpose" this kind of decomposition to a 2-connected outerplanar graph G. Intuitively, P^* corresponds to a "path of faces" H in G (induced by the vertices of the faces of P^*) and the "branches" of $G^* \backslash P^*$ correspond to the connected components of $G \backslash H$. We show that there exists an "almost" optimal (of length at most $p\ell(G)+1$) decomposition of G which "follows" H (for a particular path of G^*) in the same way as above and that can be computed in polynomial time.

Theorem 3. *Let $G=(V,E)$ be a 2-connected outerplanar graph with n vertices. A path-decomposition of G with length at most $p\ell(G)+1$ can be computed in time $O(n^3 p\ell(G)^2)$.*

Sketch of proof : Our algorithm actually computes $O(n^2)$ decompositions, one for each pair of edges of G (since G is planar, $|E| = O(n)$), and then returns one of these decompositions of minimum length. Precisely, for any two fixed edges $e, f \in E$, the algorithm computes a particular (see below) decomposition of G where the first bag contains e and the last bag contains f.

If $e = f$, we design a greedy algorithm that computes, in time $O(n)$, an optimal decomposition of G among all decompositions such that every bag contains both endpoints of e.

The case $e \neq f$ is more technical and needs more notations (illustrated in Figure 3). Let $P^* = (F_1, \dots, F_r)$ be the shortest path in G^* such that $e \in V(F_1)$ and $f \in V(F_r)$ (possibly $F_1 = F_r$). Let us recall that P^* corresponds to a subgraph (a "path of faces") H of G. For $1 \leq i < r$, let e_i be the edge shared by F_i and F_{i+1} ($e = e_0$ and $f = e_r$). For each component C of $G \backslash H$, let e_C be the edge of H such that $N(C) = \{u \in V \setminus C \mid \exists v \in C, \{u,v\} \in E\} = e_C$. Moreover, let D_C be an optimal decomposition of $C \cup e_C$ among all its decompositions with e_C in all their bags. Finally, for $1 \leq i \leq r$, let G_i be the subgraph of G induced by the vertices of the face F_i and the vertices of each component C of $G \setminus H$ such that $e_C \in E(F_i)$ (the set of these components is denoted by $\mathcal{C}_i$). We give an algorithm that computes a decomposition D^{ef} with minimum length among the decompositions that respect these following three properties:

1. **"face of H by face of H, from left to right (from 1 to r)" :** $D^{ef} = D_1 \odot \cdots \odot D_r$ ($\odot$ is the concatenation of sequences) such that, for $1 \leq i \leq r$, D_i is a decomposition of G_i (we decompose sequentially G_1 to G_r) such that e_{i-1} is in its first bag and e_i is in its last bag;
2. **for each face of H, "component per component" of $G \setminus H$:** for all $1 \leq i \leq r$, components of $\mathcal{C}_i$ are "added" one by one to $D_i = (X^i_1, \cdots, X^i_{r_i})$. Formally, for all $C \in \mathcal{C}_i$,

there exists $1 \leq a_C \leq b_C \leq r_i$ such that (1) the restriction of $(X^i_{a_C}, \cdots, X^i_{b_C})$ to $C \cup e_C$ is exactly D_C and (2) for all $a_C \leq j \leq j' \leq b_C$, $X^i_j \setminus V(C \cup e_C) = X^i_{j'} \setminus V(C \cup e_C) \subseteq V(F_i)$. Informally, once a vertex of a component $C \in \mathcal{C}_i$ is "added" to D_i, we cannot "add" a vertex from another component while all vertices of C have not been "added" to D_i;

3. **for each face of H, from "left to right" :** for all $1 \leq i \leq r$, the order in which components of $\mathcal{C}_i$ are "added" to D_i is restricted to the following manner (intuitively, "from left to right"). Precisely, let A_i et B_i be the two disjoint paths from e_i to e_{i+1} in F_i. For all $C, C' \in \mathcal{C}_i$ such that $e_C, e_{C'} \in E(A_i)$ (resp., $e_C, e_{C'} \in E(B_i)$) with e_C closer to e_i than $e_{C'}$ in A_i (resp., in B_i), then C has to "be" in D_i before C' (i.e., $b_C < a_{C'}$).

Once the two edges e and f ($e \neq f$) are fixed, our algorithm computes first the decomposition D_C for every component C of $G \setminus H$, which can be done in global time $O(n)$ by the greedy algorithm previously mentioned. Then, for each $1 \leq i \leq r$, we compute an optimal (that minimizes the length) order (property 3) using a dynamic programming algorithm which computes such an order, and so D_i, in times $O(|F_i|^2) = O(p\ell(G)^2)$ (each inner face F is an isometric cycle and so $|F| = O(p\ell(G))$). Hence, our algorithm computes D^{ef} in time $O(n + r \cdot p\ell(G)^2) = O(n \cdot p\ell(G)^2)$.

Therefore, the $O(n^2)$ decompositions can be computed in time $O(n^3 p\ell(G)^2)$. The key point is to prove that there exists such a path-decomposition with length at most $p\ell(G) + 1$. We show that, for any optimal decomposition D of G, if there is no edge contained in each bag, then we can modify D in a decomposition of G satisfying the 3 properties (for two different edges $e, f \in E$ depending on D) with length at most $p\ell(G) + 1$. $\diamond$

Further work. The next step will be to design a polynomial time exact algorithm (if it exists) to compute the pathlength of outerplanar graphs. Note that there exist graphs such that all previous particular decompositions have length at least $p\ell(G) + 1$ (see Fig. 2), and that this raise is due to the second property presented above. Note also that we considered 2-connected outerplanar graphs to simplify the presentation (our results extend to any outerplanar graph). Another interesting question is to know if our algorithm for trees can be adapted in the case of chordal graphs. Finally, the complexity of computing the pathlength (treelength) of planar graphs is still open.

References

[1] R. Belmonte, F. V. Fomin, P. A. Golovach and M. S. Ramanujan. Metric Dimension of Bounded Tree-length Graphs. SIAM J. Discret. Math., 32(2): 1217-1243, 2017.

[2] H. L. Bodlaender and F. V. Fomin. Approximation of pathwidth of outerplanar graphs. J. Alg., 43(2): 190-200, 2002.

[3] D. Coudert, F. Huc and J.-S. Sereni. Pathwidth of outerplanar graphs. J. Graph Theory, 55(1): 27-41, 2007.

[4] T. Dissaux, G. Ducoffe, N. Nisse and S. Nivelle. Treelength of Series-parallel Graphs. In XI Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS), 195(1): 30-38, 2021.

[5] T. Dissaux and N. Nisse. Pathlength of Outerplanar graphs. Technical Report, http://www-sop.inria.fr/members/Thomas.Dissaux/Outerplanar.pdf.

[6] Y. Dourisboure and C. Gavoille. Tree-decompositions with bags of small diameter. Discret. Math., 307(16): 2008-2026, 2007.

[7] F. F. Dragan, E. Köhler and A. Leitert. Line-Distortion, Bandwidth and Path-Length of a Graph. Algorithmica, 77(3): 686-713, 2017.

[8] G. Ducoffe, S. Legay and N. Nisse. On the Complexity of Computing Treebreadth. Algorithmica, 82(6): 1574-1600, 2020.

[9] R. Krauthgamer and J. R. Lee. Algorithms on negatively curved spaces. In 47th Annual IEEE Symp. on Foundations of Computer Science (FOCS 2006): 119-132, 2006.