



HAL
open science

Pathlength of Outerplanar graphs

Thomas Dissaux, Nicolas Nisse

► **To cite this version:**

Thomas Dissaux, Nicolas Nisse. Pathlength of Outerplanar graphs. [Research Report] Inria & Université Nice Sophia Antipolis, CNRS, I3S, Sophia Antipolis, France. 2022. hal-03655637v2

HAL Id: hal-03655637

<https://hal.science/hal-03655637v2>

Submitted on 10 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pathlength of Outerplanar graphs*

Thomas Dissaux¹ and Nicolas Nisse¹

¹Université Côte d’Azur, CNRS, Inria, I3S, Sophia Antipolis, 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 $pl(G)$, of G is the smallest length of its path-decompositions. This parameter has been studied for its algorithmic applications for several classical metric problems like the minimum eccentricity shortest path problem, the line-distortion problem, *etc.* However, deciding if the pathlength of a graph G is at most 2 is NP-complete, and the best known approximation algorithm has a ratio 2 (there is no c -approximation with $c < \frac{3}{2}$ unless $P = NP$). In this work, we focus on the study of the pathlength of simple sub-classes of planar graphs. We start by designing a linear-time algorithm that computes the pathlength of trees. Then, we show that the pathlength of cycles with n vertices is equal to $\lfloor \frac{n}{2} \rfloor$. Finally, our main result is a (+1)-approximation algorithm for the pathlength of outerplanar graphs. This algorithm is based on a characterization of almost optimal (of length at most $pl(G) + 1$) path-decompositions of outerplanar graphs.

1 Introduction

Path-decompositions of graphs have been extensively studied since their introduction in the Graph Minor theory by Robertson and Seymour, for their various algorithmic applications. A *path-decomposition* of a graph $G = (V, E)$ is a sequence (X_1, \dots, X_p) of subsets (called *bags*) of V such that (1) $\bigcup_{i \leq p} X_i = V$, (2) for all edges $\{u, v\} \in E$, there exists $1 \leq i \leq p$ such that $u, v \in X_i$, and (3) for all $1 \leq i \leq z \leq j \leq p$, $X_i \cap X_j \subseteq X_z$. These constraints imply this fundamental property (widely used in proofs), for all $1 \leq i < p$, $S = X_i \cap X_{i+1}$ *separates* $A = \bigcup_{j \leq i} X_j \setminus S$ and $B = \bigcup_{j \geq i} X_j \setminus S$ (i.e. every path between A and B goes through S).

The most classical measure of path-decompositions is their *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 theorem of Courcelle implies that numerous NP-hard problems can be solved in polynomial times in graphs of bounded pathwidth [2].

We focus on an other measure of path-decomposition which, while less studied, has also numerous algorithmic applications. This measure, *the length* $\ell(D)$ of a path-decomposition D , is the maximum diameter of the bags of D , where the diameter $\ell(X)$ of a bag X is the largest distance (in G) between two vertices of X . The *pathlength* $pl(G)$ of a graph G , is the minimum length of its path-decompositions [5]. In particular, this measure captures several metric properties of graphs. For example, the line distortion problem can be approximated (up to a constant factor) when the pathlength is bounded by a constant [6], which has many

*This work is partially funded by the project UCA JEDI (ANR-15-IDEX-01) and EUR DS4H Investments in the Future (ANR-17-EURE-004).

applications in computer vision [15], computational chemistry and biology [11], in network design and distributed protocol [9], *etc.* Moreover, since the pathlength is an upper bound or the treelength, the Traveling Salesman Problem admits a FPTAS in bounded pathlength graphs [13], efficient compact routing schemes and sparse additive spanners can be built in the class of graphs with bounded pathlength [12], and computing the *metric dimension* is FPT in the pathlength plus the maximum degree [1], *etc.*

Unfortunately, deciding if the pathlength of a graph is at most 2 ($pl(G) \leq 2$) is NP-complete and there does not exist a c -approximation for any $c < \frac{3}{2}$ (unless $P = NP$) [8]. On the other hand, there exists a 2-approximation in general graphs [6]. While computing the pathwidth of planar graphs is known to be NP-complete [14], the case of pathlength has not been studied yet. In this paper, we initiate this study by considering outerplanar graphs. Note that the pathwidth of outerplanar graphs is known to be polynomial-time solvable, but the best known algorithm to compute the pathwidth of outerplanar n -node graphs has complexity at least $O(n^{11})$ [3]. Moreover, there exist 2-approximations for this problem, with time complexity $O(n \log(n))$, that deal with the problem by relating the pathwidth of an outerplanar graph with the one of its weak dual [3, 4].

Our contributions. In Section 3, we first present a linear-time algorithm that computes the pathlength of trees, and prove that $pl(C_n) = \lfloor \frac{n}{2} \rfloor$ for any cycle C_n with n vertices. Section 4 is devoted to our main contribution. We design an algorithm that computes, in time $O(n^3(n + pl(G)^2))$, a path-decomposition of length at most $pl(G) + 1$ for any outerplanar n -node graph G . This algorithm is based on a structural characterization of almost optimal (of length at most $pl(G) + 1$) path-decompositions of outerplanar graphs.

2 Preliminaries

Let $G = (V, E)$ be any graph. When it will not be specified below, n will always be the number $|V|$ of vertices. In what follows, any edge $\{x, y\}$ is also considered as the set of two vertices x and y . In particular, we say that $X \subseteq V$ contains an edge e if $e \subseteq X$. Given a vertex $v \in V$, let $N(v) = \{u \in V \mid \{v, u\} \in E\}$ be the neighbourhood of v and let $N[v] = N(v) \cup \{v\}$ be its *closed neighbourhood*. Given $S \subseteq V$, let $N(S) = \{v \in V \setminus S \mid \exists u \in S, \{u, v\} \in E\}$ and let $G[S] = (S, E \cap (S \times S))$ be the subgraph of G induced by the vertices of S . The *distance* $dist_G(u, v)$ (or $dist(u, v)$ if there is no ambiguity) between $u \in V$ and $v \in V$ is the minimum length (number of edges) of a path between u and v in G . A subgraph H of G is *isometric* if $dist_H(u, v) = dist_G(u, v)$ for all $u, v \in V(H)$.

A *path-decomposition* of G is a sequence $D = (X_1, \dots, X_p)$ of subsets of vertices, called *bags*, such that $\bigcup_{i \leq p} X_i = V$, for every $e \in E$, there exists $i \leq p$ with $e \subseteq X_i$ and, for every $1 \leq i \leq j \leq q \leq p$, $X_i \cap X_q \subseteq X_j$. The *length* $\ell(D)$ of D is the maximum diameter of one of its bags, i.e., $\ell(D) = \max_{i \leq p} \ell(X_i) = \max_{i \leq p} \max_{u, v \in X_i} dist_G(u, v)$. The *pathlength* $pl(G)$ of G is the minimum length of its path-decompositions. A path-decomposition of G of length $pl(G)$ is said *optimal*.

Let us say that a path-decomposition is *reduced* if no bag is contained in another one. It is easy to check that any graph admits an optimal reduced path-decomposition. In what follows, we will use the following result.

Lemma 1 [5] *For every isometric subgraph H of G , then $pl(H) \leq pl(G)$.*

An alternative (equivalent) way to define path-decompositions of a graph $G = (V, E)$ is through orderings of V . This representation will be particularly useful in Section 4. Let $\mathcal{O} =$

(v_1, \dots, v_n) be an ordering (i.e., a permutation) of V . Let $D(\mathcal{O}) = (X_1, \dots, X_{2n-2})$ be defined as follows. Let $X_1 = \{v_1\}$ and, for every $2 \leq i < n$, let $X_{2i-2} = X_{2i-3} \cup \{v_i\}$, let $S_i \subseteq X_{2i-2}$ be the set of vertices $x \in X_{2i-2}$ such that $N[x] \subseteq \bigcup_{j \leq 2i-2} X_j$ (i.e., all vertices of X_{2i-2} whose all neighbours belong to a bag X_j for some $j \leq 2i-2$) and let $X_{2i-1} = X_{2i-2} \setminus S_i$. In particular, note that $S_i \subseteq N[v_i]$ and may be empty. Finally, let $X_{2n-2} = X_{2n-3} \cup \{v_n\}$. The following claim is straightforward.

Claim 1 *Let \mathcal{O} be any ordering of the vertex-set V of a graph $G = (V, E)$. Then, $D(\mathcal{O})$ is a (not reduced) path-decomposition of G .*

Given an ordering $\mathcal{O} = (v_1, \dots, v_n)$ and $1 \leq i \leq j \leq n$, let $\mathcal{O}[v_i, v_j] = (v_i, \dots, v_j)$.

Given two sequences $D = (X_1, \dots, X_p)$ and D' of subsets of V and $S \subseteq V$, let $D \cup S = (X_1 \cup S, \dots, X_p \cup S)$ (If $S = \{v\}$, we note $D \cup v$ instead of $D \cup \{v\}$). Let $D \cap S$ and $D \setminus S$ be defined in a similar way (in these cases, the empty bags that may be created are removed). Finally, let $D \odot D'$ be the sequence obtained by concatenation of D and D' . The following claim is straightforward (and well known).

Claim 2 *Let D be a path-decomposition of $G = (V, E)$ and $S \subseteq V$.*

1. *Then, $D' = D \cap S$ (resp., $D' = D \setminus S$) is a path-decomposition of $G[V \cap S]$ (resp., of $G \setminus S$). Moreover, if $G[V \cap S]$ (resp., $G \setminus S$) is an isometric subgraph of G , then $\ell(D') \leq \ell(D)$.*
2. *Let D' be a path-decomposition of $G[V \setminus S]$. Then, $D' \cup S$ is a path-decomposition of G .*
3. *Let $V = A \cup B$ with $A \cap B = S$ and S separating $A \setminus S$ and $B \setminus S$ (there does not exist any edge $\{u, v\} \in E$ with $u \in A \setminus S$ and $v \in B \setminus S$), let D_1 be a path-decomposition of $G[A]$ with last bag containing S , and let D_2 be a path-decomposition of $G[B]$ with the first bag containing S . Then, $D_1 \odot D_2$ is a path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\}$.*

3 Pathlength of trees and cycles

We first begin by the trees, the easiest sub-class of planar graphs. Note that, intuitively, the algorithm presented in Section 4, that computes the pathlength of outerplanar graphs follows similar ideas as the one we present for trees.

Let $k \in \mathbb{N}$ and let S_k be the tree obtained from a star with three leaves by subdividing each edges k times, i.e., by replacing each edge by a path with $k+1$ edges.

Lemma 2 [5] *For any $k \in \mathbb{N}$, $p\ell(S_k) = k+1$.*

Note that, $p\ell(S_k) = \frac{|V(S_k)|-1}{3}$. This is in contrast to the width counterpart where the pathwidth of any n -node tree is at most $O(\log n)$.

Theorem 1 *The pathlength of any tree and an optimal path-decomposition can be computed in linear time.*

Proof. Let T be any tree. If T is a path, the result is obvious (the pathlength equals one), so we may assume that T has at least three leaves.

Let P_0 be a longest path in T and let x and y be its endpoints (that are leaves). Let z be any leaf of T maximizing its distance to P_0 (i.e., for any leaf f of T , $\text{dist}(f, P_0) = \min_{v \in V(P_0)} \text{dist}(f, v) \leq \text{dist}(z, P_0)$). Let v^* be the projection of f on P_0 , i.e., v^* is the vertex of P_0 minimizing its distance to z . Note that $\text{dist}(v^*, f) \leq \min\{\text{dist}(v^*, x), \text{dist}(v^*, y)\}$

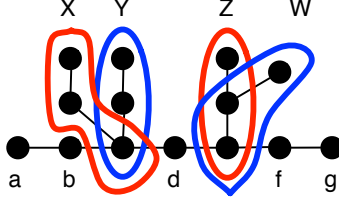


Figure 1: example of a tree T (with $pl(T) = 2$) and the path-decomposition obtained from theorem 1 could be $(\{ab\}, \{bc\}, X, Y, \{cd\}, \{de\}, Z, W, \{ef\}, \{fg\})$.

(since otherwise, P_0 would not be a longest path). Let T' be the inclusion-minimal subtree of T containing x, y and z . Note that T' contains S_k as isometric subgraph for $k = dist(v^*, z) - 1$, and so, S_k is an isometric subgraph of T . Then, by Lemmas 2 and 1, $pl(T) \geq pl(S_k) = k + 1$.

We now show that $pl(T) = k + 1$ by building a path-decomposition D of T with length $k + 1$. Let $P_0 = (x = v_0, v_1, \dots, v_t = y)$. Let us describe an iterative algorithm that builds D . Let us start with D containing only the bag $\{v_0, v_1\}$.

For i going from 1 to $t - 1$, let us do the following. Let T^i be the connected component of $T \setminus E(P_0)$ (that is, removing the edges of P_0 but keeping its vertices) that contains v_i . Let F^i be the sequence of leaves of T^i except v_i (if v_i is a leaf of T^i) ordered by following any DFS of T^i starting from v_i . Note that, by definition of z , for every $f \in F^i$, $dist(f, v_i) \leq k + 1$ (let us denote this Property by (**)). Let $\mathcal{Q} = (Q_1, \dots, Q_{|F^i|})$ be the sequence of the paths from v_i to f , for every $f \in F^i$, where the ordering of the paths follows the ordering of the leaves in F^i . For q going from 1 to $|F^i|$, add $V(Q_q)$ as the next bag of D , and then, add the bag $\{v_i, v_{i+1}\}$ to D .

Let $D = (X_1, \dots, X_h)$ be the outcome of the above algorithm. First, by Property (**), for every $1 \leq i \leq h$, $diam(X_i) = \max_{u, v \in X_i} dist_G(u, v) \leq k + 1$. Hence, it remains to prove that D is a path-decomposition of T . By construction, it is clear that every vertex and every edge of T belong to at least one bag of D . Moreover, for every $1 \leq j \leq t$, again by construction, if $v_j \in X_a \cap X_b$, then $v_j \in X_c$ for every $a \leq c \leq b$. Finally, let $v \in V(T) \setminus V(P_0)$, let $1 \leq i < t$ such that $v \in V(T^i)$ and assume that v belongs to $X_a \cap X_b$ for $1 \leq a < b \leq h$. By construction, there are two leaves f_a and f_b of $T^i \setminus \{v_i\}$ such that v belongs to P_a and P_b , the path between v_i and f_a and the path between v_i and f_b respectively, and with $X_a = V(P_a)$ and $X_b = V(P_b)$. Moreover, $f_a < f_b$ in the ordering of F^i . Now, for every $a \leq c \leq b$, the bag X_c corresponds to the path P_c between v_i and a leaf f_c of $T^i \setminus \{v_i\}$ such that $f_a \leq f_c \leq f_b$ in the ordering of F^i . By definition of the DFS of T^i from v_i , the path P_c must also contain v and so $v \in X_c = V(P_c)$. Hence, D is a path-decomposition of T with length at most $k + 1$. An example of such a decomposition is given in Figure 1.

The fact that the above algorithm can be done in linear time comes from the fact that P_0 can be computed in linear time (the diameter of a tree can be computed using 2 BFSs) and then, performing one DFS from v_i for each T^i suffices to compute the order of Q for each T^i . ■

Remark. The above proof actually shows that, in any tree T , $pl(T)$ equals its minimum eccentricity shortest-path. Note that it was already known that the minimum eccentricity shortest-path of trees can be computed in linear time [7].

We will focus now on cycles.

Theorem 2 *Let C_n be a cycle of length n . We have that $pl(C_n) = \lfloor \frac{n}{2} \rfloor$.*

Proof. First, the path-decomposition consisting of a unique bag $V(C_n)$ has length $\lfloor \frac{n}{2} \rfloor$, so, $pl(C_n) \leq \lfloor \frac{n}{2} \rfloor$. Let us show that it is an optimal path-decomposition.

For purpose of contradiction, let us suppose that C_n admits a reduced path-decomposition $D = (X_1, \dots, X_p)$ of length $k < \lfloor \frac{n}{2} \rfloor$. First, let us show that we may assume that X_1 induces a connected subgraph (more precisely, a sub-path). Let us assume that it is not the case. Since D is reduced, there is a vertex in X_1 that is not in every X_i for $2 \leq i \leq p$. Let v be such a vertex. Since v is only in X_1 , the two neighbors of v also belong to X_1 . Let A be the connected component of $G[X_1]$ containing v . Note that A is a path, otherwise $V(A) = V(C_n)$ contradicting that $k < \lfloor \frac{n}{2} \rfloor$. Let x and y be the two endpoints of A and let $A' = A \setminus \{x, y\}$. Then, $D' = (A, X_1 \setminus A', X_2 \setminus A', \dots, X_p \setminus A')$ is a path-decomposition of length at most k and whose first bag induces a path. Up to removing the redundant bags (note that A is not contained in any other bag), we may assume that D' is reduced.

Hence, let $D = (X_1, \dots, X_p)$ be a reduced path-decomposition of length $k < \lfloor \frac{n}{2} \rfloor$ such that X_1 induces a path. We moreover assume that, among such decompositions, D maximizes $|X_1|$. We next show that $|X_1| = k + 1$. Let x and y be the two ends of the path induced by X_1 . Note that $|X_1| - 1 = \text{dist}(x, y) \leq k$. For purpose of contradiction, let us assume that $\text{dist}(x, y) < k$. Let x' and y' be respectively the neighbors of x and y not in X_1 . Let X_j and $X_{j'}$ be the first bags (closest to X_1) that contain y' and x' respectively. Note that $y \in X_i$ for any $1 \leq i \leq j$ (resp. $x \in X_{i'}$ for any $1 \leq i' \leq j'$), otherwise there would be no bag containing the edge $\{y, y'\}$ (resp. the edge $\{x, x'\}$). W.l.o.g., let suppose that $j \leq j'$. Therefore, X_j contains x, y and y' . Note that $\text{dist}(x, y') = \text{dist}(x, y) + 1 \leq k$. Then, $D' = (X_1 \cup \{y'\}, \dots, X_{j-1} \cup \{y'\}, X_j, \dots, X_p)$ is a path-decomposition whose first bag induces a path of length $|X_1|$. We show that D' has length at most k which will contradict the maximality of $|X_1|$ in D .

For purpose of contradiction, let us assume that D' has length strictly larger than k . Therefore, there exists $1 < h < j$ and $w \in X_h$ such that $\text{dist}(w, y') > k$. Since $y \in X_h$, then $\text{dist}(w, y) \leq k$. Note also that $\text{dist}(w, y') \leq \text{dist}(w, y) + 1 \leq k + 1$ and so $\text{dist}(w, y') = k + 1$. Since $\text{dist}(w, y) \geq \text{dist}(w, y') - 1 = k$, then w is the unique vertex at distance k from y and at distance at least $k + 1$ from y' . Let w' be the neighbor of w such that $\text{dist}(w', y) = k + 1$. Note that w' cannot be in a bag containing y since they are at distance $k + 1$. Therefore, it is not in X_i for $1 \leq i \leq j$ (since these bags contain y). Note that w' and w have to be in some common bag of D (since they are adjacent). Therefore, there is a bag X_l with $j < l \leq p$ that contains w and w' . Since $w \in X_h \cap X_l$, then $w \in X_j$ which contradicts the fact that P has length at most k (since $\text{dist}(w, y') > k$). Therefore, D' is a path-decomposition of length k .

Hence, let $D = (X_1, \dots, X_p)$ be a reduced path-decomposition of length $k < \lfloor \frac{n}{2} \rfloor$ such that X_1 induces a path of length exactly k . Let x' and y' be the neighbors of x and y not in X_1 . Let X_j and $X_{j'}$ be the first bags (closest to X_1) that contain y' and x' respectively. Note that $y \in X_i$ for any $1 \leq i \leq j$ (resp. $x \in X_i$ for any $1 \leq i \leq j'$). W.l.o.g., let suppose that $j \leq j'$. Therefore, X_j contains x, y and y' , a contradiction since x and y' are at distance $k + 1$ because $k < \lfloor \frac{n}{2} \rfloor$. ■

4 Outerplanar graphs

This section is devoted to our main result: a polynomial-time algorithm for computing a path-decomposition of any simple (without parallel edges nor loops) outerplanar graph G with length at most $pl(G) + 1$.

A graph $G = (V, E)$ is *outerplanar* if it can be embedded in the plane without crossing edges and such that all vertices lie on the outer face (the unbounded face). An edge of an outerplanar graph is called an *internal* edge if it does not lie on the outer face. Note that, since

we only consider simple graphs, the fact that an edge is internal or not does not depend on the outerplanar embedding. Let $E_{int} \subseteq E$ be the set of internal edges and $E_{out} = E \setminus E_{int}$ be the set of *outer* edges. Note that any internal edge $e \in E_{int}$ of an outerplanar graph $G = (V, E)$ is a separator (i.e., $G \setminus V(e)$ has several connected components).

Path-decomposition with fixed first and last elements. Let $G = (V, E)$ be a connected graph and let $v \in V$ (resp., $e \in E$). A path-decomposition $D = (X_1, \dots, X_p)$ of G *starts* from v (resp., from e) if $v \in X_1$ (resp., if $e \subseteq X_1$). Similarly, D *finishes* with v (resp., with e) if $v \in X_p$ (resp., if $e \subseteq X_p$). Let $x, y \in V \cup E$, a path-decomposition of G starting from x and finishing with y is called a $\{x, y\}$ -*path-decomposition*. Let $pl(G, x, y)$ be the minimum length among all $\{x, y\}$ -path-decompositions of G . An $\{x, y\}$ -path-decomposition is an *optimal* $\{x, y\}$ -path-decomposition if its length is $pl(G, x, y)$. Clearly, any $\{x, y\}$ -path-decomposition (X_1, \dots, X_p) corresponds to an $\{y, x\}$ -path-decomposition (X_p, \dots, X_1) of same length, and so:

Claim 3 For any connected graph $G = (V, E)$ and $x, y \in E \cup V$, $pl(G, x, y) = pl(G, y, x)$.

The following claim directly holds by definition and because there always exists a reduced optimal path-decomposition (note that, for any reduced path-decomposition D of a connected graph, there exist $x, y \in E$ such that D starts from x and finishes with y).

Claim 4 For any connected graph $G = (V, E)$, $pl(G) = \min_{x, y \in E} pl(G, x, y)$.

For our purpose, we need to refine the above claim as follows.

Lemma 3 For any connected graph $G = (V, E)$, $pl(G) = \min_{x, y \in E_{out}} pl(G, x, y)$.

Proof. Let $x, y \in E$ such that $pl(G) = pl(G, x, y)$ and such that the distance between x and y is maximized (here, the distance can be viewed at the number of internal faces that must be crossed to go from x to y). Let $D = (X_1, \dots, X_p)$ be an optimal $\{x, y\}$ -path-decomposition of G (so D is an optimal path-decomposition of G).

For purpose of contradiction, let us assume that $x \in E_{int}$. Let C be a connected component of $G \setminus x$ that does not contain y . Let $D' = D \cap (C \cup x)$ (note that $\ell(D') \leq \ell(D)$ by Claim 2 and because $G \cap (C \cup x)$ is an isometric subgraph of G) and $D_2 = D \setminus C$ (note that $\ell(D_2) \leq \ell(D)$ by Claim 2 and because $G \setminus C$ is an isometric subgraph of G). Let D'' be the reduced path-decomposition of $G[C \cup x]$ obtained from D' (i.e., by removing the bags that are contained in others). Since D'' is reduced, the last bag of D'' must contain an edge $x' \neq x$ of $G[C \cup x]$. Finally, let D_1 be the path-decomposition of $G[C \cup x]$ obtained from D'' by reversing the order of the bags. Then, $D_1 \odot D_2$ is a $\{x', y\}$ -path-decomposition of G of length at most $pl(G)$ (by Claim 2 and since D_1 and D_2 contains x respectively in its last bag and in its first bag), contradicting the maximality of the distance between x and y . ■

By lemma 3, the computation of $pl(G)$ can be restricted to the $O(n^2)$ computations of $pl(G, x, y)$ for all fixed $x, y \in E_{out}$ (since G is planar, $|E| = O(n)$). Most of what follows is devoted to this task. Therefore, in Sections 4.1 to 4.2, $x, y \in V \cup E_{out}$ will be fixed. Section 4.1 is devoted to the “easy” cases. Depending on x and y , it will sometimes be possible to reduce the problem to smaller instances (and then proceed by induction) and, in other cases (roughly when $x = y$ and $G \setminus x$ is connected), we present a greedy algorithm that computes an optimal $\{x, y\}$ -path-decomposition of G . Section 4.2 is devoted to the remaining case: roughly, when x and y are distinct, not included in each-other, do not separate G and belong to a same internal (bounded) face of G . In this latter case, we show that we can restrict our attention to particular $\{x, y\}$ -path-decompositions of G and that such an optimal decomposition can be computed in polynomial time by dynamic programming. Finally, Section 4.3 formally states

our main result and describes our main algorithm that uses results of Sections 4.1 to 4.2 to compute a path-decomposition of a connected outerplanar graph with length at most $p\ell(G) + 1$.

4.1 Cases (depending on x and y) when recursion or greedy algorithm are possible

Let $G = (V, E)$ be a connected simple outerplanar graph and let $x, y \in E_{out} \cup V$. In this section, we first show that if $G \setminus x$ is not connected (Section 4.1.1) or if x and y are “separated” (see formal definition in Section 4.1.2), the problem of computing an optimal $\{x, y\}$ -path-decomposition of G can be reduced to similar problems in smaller instances. Then, in subsection 4.1.3, roughly, we show that, if none of the previous two properties is met and either $x = y$ or $x \in y$ or $y \in x$, then an optimal $\{x, y\}$ -path-decomposition of G can be computed in linear time by a greedy algorithm.

4.1.1 Case when $G \setminus x$ is not connected

Let us first assume that $G \setminus x$ is not connected. We have divided this case in several lemmas that look very similar (proofs seem redundant). We have not been able to factorize them while keeping a reasonable readability of this part. In particular, we hope that the division in different lemmas will be helpful to convince that all cases are actually considered.

Lemma 4 *Let $G = (V, E)$ be a simple connected outerplanar graph and let $x = y \in E_{out} \cup V$ such that $G \setminus x$ is not connected.*

Let C be any connected component of $G \setminus x$, let D_1 be an optimal $\{x, x\}$ -path-decomposition of $G[C \cup x]$ and let D_2 be an optimal $\{x, x\}$ -path-decomposition of $G \setminus C$. Then, $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let D be an optimal $\{x, x\}$ -path-decomposition of G . Then, since $C \cup \{x\}$ and $G \setminus C$ are isometric subgraphs of G and by Claim 2, $D \cap (C \cup \{x\})$ is a $\{x, x\}$ -path-decomposition of $G[C \cup x]$ of length at most $\ell(D)$ and $D \setminus C$ is a $\{x, x\}$ -path-decomposition of $G \setminus C$ of length at most $\ell(D)$.

Reciprocally, by the last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. ■

Therefore, in the remaining part of this subsection (i.e., the case where $G \setminus x$ is not connected), we may moreover assume that $x \neq y$.

In the next four lemmas, we consider the cases when $x = \{u, v\} \in E_{out}$, $G \setminus x$ is not connected and $x \neq y$ ($y \in V \cup E_{out}$). Note that, because $x = \{u, v\}$ is not an internal edge, the fact that $G \setminus x$ is not connected implies that one (or both) vertex of $\{u, v\}$ is a cut-vertex.

Precisely, w.l.o.g., we assume that $G \setminus u$ is not connected, and: in Lemmas 5 and 6, we assume that $x \cap y = \{u\}$, and in Lemmas 7 and 8, we consider the cases when $u \notin x \cap y$.

Lemma 5 *Let $G = (V, E)$ be a simple connected outerplanar graph and let $x = \{u, v\} \in E_{out}$ such that $G \setminus u$ is not connected and $\{u\} = x \cap y$. Let C be the connected component of $G \setminus u$ that contains v and assume that y does not intersect C .*

Let D_1 be an optimal $\{x, u\}$ -path-decomposition of $G[C \cup x]$ and D_2 be an optimal $\{u, y\}$ -path-decomposition of $G \setminus C$, then $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let D be an optimal $\{x, y\}$ -path-decomposition of G (note that u belongs to all bags of D). By first item of Claim 2 and because $C \cup x$ and $G \setminus C$ are isometric subgraph of G ,

$D_1 = D \cap (C \cup x)$ is a $\{x, u\}$ -path-decomposition of $G[C \cup x]$ of length at most $p\ell(G, x, y)$ and $D_2 = D \setminus C$ is a $\{u, y\}$ -path-decomposition of $G \setminus C$ of length at most $p\ell(G, x, y)$.

Reciprocally, by the last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then clearly, $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. ■

Lemma 6 *Let $G = (V, E)$ be a simple connected outerplanar graph and let $x = \{u, v\} \in E_{out}$ such that $G \setminus u$ is not connected and $\{u\} = x \cap y$. Assume that y intersects the connected component C' of $G \setminus u$ that contains v and let C be any connected component of $G \setminus u$ that does not contain v .*

Let D_1 be an optimal $\{x, x\}$ -path-decomposition of $G[C \cup x]$ and D_2 be an optimal $\{x, y\}$ -path-decomposition of $G \setminus C$, then $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let D be an optimal $\{x, y\}$ -path-decomposition of G (note that u belongs to all bags of D). By Claim 2 and because $C \cup u$ and $G \setminus C$ are isometric subgraph of G , $D_1 = (D \cap (C \cup u)) \cup v$ is a $\{x, x\}$ -path-decomposition of $G[C \cup x]$ of length at most $p\ell(G, x, y)$ (we can add v since all bags of D contain a vertex of C' , i.e., a vertex between v and $y \cap C'$) and $D_2 = D \setminus C$ is a $\{x, y\}$ -path-decomposition of $G \setminus C$ of length at most $p\ell(G, x, y)$.

Reciprocally, by the last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then clearly, $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. ■

Lemma 7 *Let $G = (V, E)$ be a simple connected outerplanar graph and let $x = \{u, v\} \in E_{out}$ such that $G \setminus u$ is not connected and $u \notin x \cap y$. Let C be the component of $G \setminus u$ that contains v and assume that C does not intersect y .*

Let D_1 be an optimal $\{x, u\}$ -path-decomposition of $G[C \cup x]$ and D_2 be an optimal $\{u, y\}$ -path-decomposition of $G \setminus C$, then $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let D be an optimal $\{x, y\}$ -path-decomposition of G . Then, by Claim 2 and because $C \cup u$ and $G \setminus C$ are isometric subgraph of G , $D_1 = (D \cap (C \cup u)) \cup u$ is a $\{x, u\}$ -path-decomposition of $G[C \cup x]$ of length at most $p\ell(G, x, y)$ (this holds since every bag of D contains u or a vertex of the component of $G \setminus u$ that contains y) and $D_2 = D \setminus C$ is a $\{u, y\}$ -path-decomposition of $G \setminus C$ of length at most $p\ell(G, x, y)$.

Reciprocally, by the last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then clearly, $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. ■

Lemma 8 *Let $G = (V, E)$ be a simple connected outerplanar graph and let $x = \{u, v\} \in E_{out}$ such that $G \setminus u$ is not connected and $u \notin x \cap y$. Assume that y is contained in the component C' of $G \setminus u$ that contains v . Let C be any connected component of $G \setminus u$ that does not contain y .*

Let D_1 be an optimal $\{x, x\}$ -path-decomposition of $G[C \cup x]$ and D_2 be an optimal $\{x, y\}$ -path-decomposition of $G \setminus C$, then $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let D be an optimal $\{x, y\}$ -path-decomposition of G . Then, by Claim 2 and because $C \cup u$ and $G \setminus C$ are isometric subgraph of G , $D_1 = (D \cap (C \cup u)) \cup x$ is a $\{x, x\}$ -path-decomposition of $G[C \cup x]$ of length at most $p\ell(G, x, y)$ (this holds since every bag of D contains a vertex of C' between v and $y \cap C'$ and a vertex of C' between u and $y \cap C'$) and $D_2 = D \setminus C$ is a $\{x, y\}$ -path-decomposition of $G \setminus C$ of length at most $p\ell(G, x, y)$.

Reciprocally, by the last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then clearly, $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. ■

The case when $x \in V$, $y \in E_{out}$ and $x \in y$ can be dealt similarly by reversing the path-decomposition (Claim 3).

It only remains the case when $x, y \in V$, $x \neq y$ and $G \setminus x$ is not connected.

Lemma 9 *Let $G = (V, E)$ be a simple connected outerplanar graph and let $x, y \in V$ such that $G \setminus x$ is not connected and $y \neq x$. Let C be any connected component of $G \setminus x$ that does not contain y .*

Let D_1 be an optimal $\{x, x\}$ -path-decomposition of $G[C \cup x]$ and D_2 be an optimal $\{x, y\}$ -path-decomposition of $G \setminus C$, then $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let D be an optimal $\{x, y\}$ -path-decomposition of G . Then, by Claim 2 and because $C \cup x$ and $G \setminus C$ are isometric subgraphs of G , $D_1 = D \cap (C \cup x) \cup x$ is a $\{x, x\}$ -path-decomposition of $G[C \cup x]$ of length at most $p\ell(G, x, y)$ (this holds since every bag of D contains x or a vertex in $V \setminus C$ since y is in the last bag) and $D_2 = D \setminus C$ is a $\{x, y\}$ -path-decomposition of $G \setminus C$ of length at most $p\ell(G, x, y)$.

Reciprocally, by the last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then clearly, $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. ■

From now on, we may assume that $G \setminus x$ is connected. For the same reasons and by Claim 3 (and the sentence preceding this claim), we may also assume that, $G \setminus y$ is connected.

4.1.2 Case when x and y are “separated”

Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$. In this section, we always assume that $G \setminus x$ and $G \setminus y$ are connected. We say that x and y are *separated* in G if there exists $z \in V \cup E$ such that $x' = x \setminus z \neq \emptyset$, $y' = y \setminus z \neq \emptyset$ and all paths from x' to y' intersect z . Note that this requires that $x \neq y$ and neither $x \in y$ nor $y \in x$, but x and y may intersect (in which case, z is or contains $x \cap y$). Note also that, if $z \in E$ and is inclusion-minimal for the property of separating x and y (i.e., no end of z separates x and y), then $z \in E_{int}$.

Lemma 10 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, and such that x and y are separated by $z \in V \cup E_{int}$.*

Let C^x be the connected component of $G \setminus z$ containing (or intersecting) x and let $C^y = V \setminus C^x$.

Let D_1 be an optimal $\{x, z\}$ -path-decomposition of $G[C^x \cup z]$ and D_2 be an optimal $\{z, y\}$ -path-decomposition of $G[C^y]$. Then, $D_1 \odot D_2$ is an optimal $\{x, y\}$ -path-decomposition of G .

Proof. Let $D = (X_1, \dots, X_p)$ be an optimal $\{x, y\}$ -path-decomposition of G . Let $1 \leq i \leq p$ be the smallest integer such that $z \in X_i$ (resp., such that $z \subseteq X_i$ if z is an edge) and let $i \leq j \leq p$ be the largest integer such that $z \in X_j$ (resp., such that $z \subseteq X_j$ if z is an edge). Then, let $D_1 = (X_1 \cap (C^x \cup z), \dots, X_j \cap (C^x \cup z), (X_{j+1} \cap (C^x \cup z)) \cup z, \dots, (X_p \cap (C^x \cup z)) \cup z)$. Clearly, D_1 is a $\{x, z\}$ -path-decomposition of $G[C^x \cup z]$. Moreover, because $y \in X_p$, then all bags X_q (for $j < q \leq p$) must contain some vertex in the connected component of $G \setminus z$ containing y . Therefore, by Claim 2, $\ell(D_1) \leq \ell(D)$. Similarly, let $D_2 = ((X_1 \setminus C) \cup z, \dots, (X_{i-1} \setminus C) \cup z, X_i \setminus C, \dots, X_p \setminus C)$. Then, D_2 is a $\{z, y\}$ -path-decomposition of $G \setminus C^x$. Moreover, because $x \in X_1$, then all bags X_q (for $1 \leq q < i$) must contain some vertex in C^x . Therefore, by Claim 2, $\ell(D_2) \leq \ell(D)$.

Reciprocally, by last statement of Claim 2, if D_1 and D_2 are defined as in the statement of the lemma, then clearly, $D_1 \odot D_2$ is an $\{x, y\}$ -path-decomposition of G of length at most $\max\{\ell(D_1), \ell(D_2)\} \leq \ell(D) = p\ell(G, x, y)$. \blacksquare

4.1.3 Case when $x = y$ or $x \in y$ or $y \in x$: greedy algorithm

In the case of this section, we show that an optimal $\{x, y\}$ -path-decomposition can be computed in linear time. We first define a recursive process to build a particular $\{x, x\}$ -path-decomposition when $x \in E_{out} \cup V$.

Let $G = (V, E)$ be a n -node connected simple outerplanar graph and $x \in E_{out}$ such that neither u nor v is a cut-vertex, or $x \in V$ and is not a cut-vertex. A *greedy path-decomposition* P of G based on x is any path-decomposition of G that can be obtained recursively (on $|V|$) as follows.

Recursive algorithm *Greedy*.

- If G is a cycle (v_1, \dots, v_n) (w.l.o.g., $x = \{v_1, v_n\}$ or $x = v_1$), $D = (X_1, \dots, X_{n-1})$ with, for every $1 \leq i \leq n-1$, $X_i = x \cup \{v_i, v_{i+1}\}$. The second basic case is when G is an edge $\{u, v\}$ and $x = u$, in which case, $D = (\{u, v\})$.
- Else, if x is a vertex of degree one in G , let w be its neighbour and D' be a greedy path-decomposition of $G \setminus x$ based on w . Then, $D = (\{x, w\}) \odot (D' \cup x)$ is a greedy path-decomposition of G based on x .
- Else, if x is a cut-vertex that belongs to no internal face of G , let C be any connected component of $G \setminus x$. Let D_1 be any greedy path-decomposition of $G[C \cup x]$ based on x and let D_2 be any greedy path-decomposition of $G \setminus C$ based on x . Then, $D = D_1 \odot D_2$ is a greedy path-decomposition of G based on x .
- Else, let (v_1, \dots, v_q) be any internal face containing x such that $x = \{v_1, v_q\}$ (resp. $x = v_1$, if x is a vertex).
 - If there exists a cut-vertex v_j for some $2 \leq j < q$ (resp. $2 \leq j \leq q$), let C be a connected component of $G \setminus v_j$ that does not contain x . Let $G_1 = G[C \cup v_j]$ and $G_2 = G[V \setminus C]$. Let D_1 be a greedy path-decomposition of G_1 based on v_j , let $D_2 = (X_1, \dots, X_p)$ be a greedy path-decomposition of G_2 based on x and let $1 \leq h \leq p$ be the largest integer such that $\{v_{j-1}, v_j\} \subseteq X_h$ (if $v_j = v_2$, let $1 \leq h \leq p$ be the largest integer such that $\{v_{j-1}, v_j\} \subseteq X_h$ minus 1). Then, $D = (X_1, \dots, X_h) \odot (D_1 \cup (X_h \cap X_{h+1})) \odot (X_{h+1}, \dots, X_p)$ is a greedy path-decomposition of G based on x .
 - Otherwise, there exists $1 \leq j < q$ (resp., $1 \leq j \leq q$) such that $f = \{v_j, v_{j+1}\} \in E_{int}$ with neither v_j nor v_{j+1} is a cut-vertex. First, if $1 \leq j < q-1$ (resp., $1 \leq j < q$) (case $j = q-1$ when x is an edge and case $j = q$ when x is a vertex are treated after), let C and C' be the two connected components of $G \setminus f$ and, w.l.o.g., C intersects x . Let D_1 be a greedy path-decomposition of $G[C' \cup f]$ based on f and let $D_2 = (X_1, \dots, X_p)$ be a greedy path-decomposition of $G[C \cup f]$ based on x and let $1 \leq h \leq p$ be the smallest integer such that $f \subseteq X_h$. Then, $D = (X_1, \dots, X_h) \odot (D_1 \cup (X_h \cap X_{h+1})) \odot (X_{h+1}, \dots, X_p)$ is a greedy path-decomposition of G based on x . Hence, it remains the case where $j = q-1$ (resp., $j = q$). Let C and C' be the two connected components of $G \setminus f$ (resp., $G \setminus \{v_q, v_1\}$) and, w.l.o.g., C intersects

x . Let $D_1 = (X'_1, \dots, X'_{p'})$ be a greedy path-decomposition of $G[C' \cup f]$ based on f and let $D_2 = (X_1, \dots, X_p)$ be a greedy path-decomposition of $G[C \cup f]$ based on x . Note that X_{p-1} and X_p contain f and x by construction.

Then, $D = (X_1, \dots, X_{p-1}) \odot (X'_1 \cup (X_{p-1} \cap X_p), \dots, X_{p'} \cup (X_{p-1} \cap X_p)) \odot (X_p)$ is a greedy path-decomposition of G based on x .

Note that, there is not a unique decomposition resulting from the above process. However, it is easy to show by induction (in particular, each edge is considered at most twice) that:

Claim 5 *Any sequence $D = (X_1, \dots, X_p)$ returned by Algorithm Greedy is a $\{x, x\}$ -path-decomposition of G . Moreover, Algorithm Greedy proceeds in linear time.*

Moreover, for all $1 < i \leq p$, $|X_i \setminus X_{i-1}| \leq 1$ and, if $X_i \setminus X_{i-1} = \{u\}$, then X_i is a subset of the set that consists of u , of one of its neighbors u' (with $\{u, u'\}$ an outer-edge), the vertices of the internal edges x belong to and of all the vertices of each internal edge or vertex that separates u from x .

Proof. The fact that the algorithm returns a path-decomposition holds by induction. It clearly holds if G is a cycle. If x has degree one (with neighbour w), then by induction, D' is a path-decomposition containing w in every bag and so $D = (\{x, w\}) \odot (D' \cup x)$ is a path-decomposition. If x is a cut-vertex, D_1 and D_2 are path-decompositions containing x in all their bags and, since $(C \cup \{x\}) \cap (V \setminus C) = \{x\}$, then $D_1 \odot D_2$ is a path-decomposition based on x . Now, let us consider the case when x is an outer-edge $\{v_1, v_q\}$ (the case when x is a vertex is similar). If there exists a cut-vertex v_j ($2 \leq j \leq q$) then, by induction, G_1 is a path-decomposition of G_1 (every vertex and edge of G_1 appears in some bag of D_1) and $G_2 = (X_1, \dots, X_p)$ is a path-decomposition of G_2 (every vertex and edge of G_1 appears in some bag of D_2). Then, since $v_j \in X_h$, $(X_1, \dots, X_h) \odot (D_1 \cup (X_h \cap X_{h+1})) \odot (X_{h+1}, \dots, X_p)$ is a path-decomposition of G (the third property being ensured since we added $X_h \cap X_{h+1}$ to all bags of D_1). The last case can be proved similarly.

The proof of the second statement is by induction. We moreover prove that $X_1 \setminus x$ consists of at most one vertex. If G is a cycle, then the result holds trivially. If x is a vertex with a unique neighbour w , then recall that $D = (\{x, w\}) \odot (D' \cup x) = (X_1, \dots, X_p)$. By induction, D' satisfies the statement and the first bag of D' contains at most one vertex in addition to w . Hence, $|X_i \setminus X_{i-1}| \leq 1$ for all $1 < i \leq p$. Moreover, if $X_i \setminus X_{i-1} = \{u\}$, since every edge of vertex that separates u from w also separates u from x , the statement holds. If x is a cut-vertex, then the statement holds by induction for D_1 and D_2 and so for $D_1 \odot D_2$. Let us finally consider the last case when $x = \{v_1, v_q\}$ (or similarly, x is a vertex) and $f = \{v_j, v_{j+1}\}$ is a cut-edge (the case when v_j is a cut-vertex is similar). Let $D = (X'_1, \dots, X'_{p'}) = (X_1, \dots, X_h) \odot (D_1 \cup (X_h \cap X_{h+1})) \odot (X_{h+1}, \dots, X_p)$ as defined in the algorithm and note that D_1 is based on f and satisfies the statement by induction. Then, the fact that $|X'_i \setminus X'_{i-1}| \leq 1$ for all $1 < i \leq p'$ clearly holds since $f \subseteq X_h$ (and the first bag of D_1 contains at most one vertex in addition to f). Let us consider the case when $X_i \setminus X_{i-1} = \{u\}$ and $u \in C'$. Since the result holds by induction for D_2 , then $X_h \cap X_{h+1}$ contains f and all vertices of each internal edge or vertex that separate f from x . Since, moreover, every edge or vertex that separates u from f also separates u from x , the statement holds. Finally, if $X_i \setminus X_{i-1} = \{u\}$ and $u \in C$, the result holds since D_2 satisfies the requirement by induction. ■

Another way to see the above path-decomposition is as follows. Let $x = \{v_1, v_n\}$ (or $x = v_1$ if x is a vertex) (v_1, \dots, v_n) be a DFS ordering of V obtained by starting from v_1 , following the outer-face of G and finishing in v_n (or going back to v_1 if x is a vertex). That is, the vertices are ordered in the order they are met along the outer-face. Note that, up to reversal, this

order is unique if G is 2-connected. Then, consider the path-decomposition $D = (X_1, \dots, X_p)$ built as follows. Let $X_1 = x$. Assume that D has been built until some bag X_i (and not all vertices have been added yet) and $X_i \setminus X_{i-1} = \{v_j\}$ ($j = 1$ if $i = 1$). Let $\{v_{j_1}, \dots, v_{j_q}\}$ be the set of neighbours of v_j such that $j < j_1 < \dots < j_q$ and $\{v_j, v_{j_h}\}$ is an internal-edge for all $1 \leq h \leq q$. Then, let $X'_i \subseteq X_i$ be the set of vertices v_s that have no neighbours in $\{v_j, \dots, v_n\}$. Then, we extend D with the bags $(X_i \setminus X'_i) \cup \{v_{j_q}\}, (X_i \setminus X'_i) \cup \{v_{j_q}, v_{j_{q-1}}\}, \dots, (X_i \setminus X'_i) \cup \{v_{j_1}, \dots, v_{j_q}\}, (X_i \setminus X'_i) \cup \{v_{j_1}, \dots, v_{j_q}, v_{j+1}\}$.

Theorem 3 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected and either $x = y$ or $y \in x$ or $x \in y$.*

An optimal $\{x, y\}$ -path-decomposition of G can be computed in linear time (in $O(|E|)$).

Proof. Let us first consider the case when $x = y$. Note that, since x belongs to every bag of every $\{x, x\}$ -path-decomposition of G , then $pl(G, x, x) \geq \max_{w \in V} dist(w, x)$ (if $x = \{u, v\} \in E$, $dist(w, x) = \max\{dist(w, u), dist(w, v)\}$).

Let $D = (X_1, \dots, X_p)$ be a result of Algorithm *Greedy*. By Claim 5, D is a $\{x, x\}$ -path-decomposition of G computed in linear time. Let $1 < i \leq p$ such that $|X_i \setminus X_{i-1}| > 0$.

Let us suppose by contradiction that there is two vertices v and v' in $X_i \setminus x$ such that $d(v', v) \geq \max_{w \in V} dist(w, x)$. Note that, v and v' cannot be both in the same edge separator of u and x , and that they cannot be both in $X_i \cap N(u)$, because in this case, $d(v, v') = 1 \leq \max_{w \in V} dist(w, x)$. For the moment, let us suppose that x is an edge. Hence, by Claim 5 and w.l.o.g., we can suppose that $v \in S_1$, a separator of x and u (resp., $S_1 = N(u)$), and $v' \in S_2$, a separator of $x \setminus S_2$ and u such that v is not in the connected component of $G \setminus S_2$ that contains x . Therefore, if S_2 is a cut vertex, then every (v, x) -path contains v' and so $d(v, v') \leq d(v, x)$, a contradiction. Otherwise, S_2 is an edge-separator (i.e. S_2 in E_{int}). Let us denote by v_x the vertex from x such that $d(v, x)$ is maximized. If $v' \in (v, v_x)$ -path, then $d(v, v') \leq d(v, v_x)$, a contradiction. Otherwise, the only vertex v^* in $S_2 \setminus v'$ is in every (v, v_x) -path and so $d(v, v') \leq d(v, v^*) + 1 \leq d(v, v_x)$ (even if $v^* \in x$), a contradiction. Note that it remains the case where x is a vertex. Note that if we can suppose that $v \in S_1$, a separator of x and u (resp., $S_1 = N(u)$), and $v' \in S_2$, a separator of $x \setminus S_2$ and u such that v is not in the connected component of $G \setminus S_2$ that contains x , then the same argument as before holds, which implies a contradiction. Otherwise, v' and x are adjacent and $G \setminus \{v', x\}$ disconnect the graph. Let us denote by $\mathcal{C}_{v'}$ the set of connected component of $G \setminus v'$ not containing x (if there is such component). Let C_1 and C_2 be the two connected component $G \setminus (\{x, v'\} \cup_{C \in \mathcal{C}_{v'}} V(C))$. Note that, by construction, however which face has been chosen, $v' = v_q$. Moreover, for any bag X containing v' , $X \cap C_1 = N_{C_1}(v')$, $X \cap C_2 = N_{C_2}(v')$. Hence it implies that $v \in C \in \mathcal{C}_{v'}$. Note that every (v, x) -path contains v' , which implies that $d(v, v') \leq d(v, x)$, a contradiction.

Therefore, the diameter $\max_{u, v \in X_i} dist_G(u, v)$ of X_i is at most $\max_{v \in V(G)} dist(v, x)$. Hence, D is an optimal $\{x, x\}$ -path-decomposition of G .

Now, let us assume that $x = \{u, v\}$ and $y = v$ (the case when $x \in y$ is symmetric). Note that, since v belongs to every bag of every $\{x, y\}$ -path-decomposition of G , then $pl(G, x, y) \geq \max_{w \in V} dist(w, v)$. Let F be the internal face containing x (such a face exists and is unique since $G \setminus x$ and $G \setminus y$ are connected, $u \in E_{out}$ and $G \neq x$) and let $w \neq v$ be the second neighbour of u in F . Let C be the connected component of $G \setminus \{u, w\}$ that does not contain v and such that $N(C) = \{u, w\}$ (possibly $C = \emptyset$). Note that, for every $h \in C$, $dist_G(h, u) \leq dist_G(h, v)$. Let $D' = (X_1, \dots, X_p)$ be a result of Algorithm *Greedy* and let $1 \leq i \leq p$ be the smallest integer such that $w \in X_i$. Let $D = (X_1, \dots, X_i, X_{i+1} \setminus \{u\}, \dots, X_p \setminus \{u\})$. Then, D is a $\{x, y\}$ -path-decomposition of G of length at most $\max_{w \in V} dist(w, v)$. ■

Note that, in the case where $y \in x$, an optimal $\{x, y\}$ path-decomposition can be computed

as a $\{x, x\}$ path-decomposition (X_1, \dots, X_q) such that, at the end of the construction, we remove $x \setminus y$ from X_{i+1}, \dots, X_q where X_i is the first bag such that $N(x) \subseteq \bigcup_{1 \leq j \leq i} X_j$. Note that by claim 3, the case $x \in y$ is similar.

4.2 Case when x and y are “around” a same face

In this section, we consider the last remaining case which is much more technical than previous ones. Namely, let $G = (V, E)$ be a n -node connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G (i.e., they are not separated by an edge or a vertex).

In this setting, we first show that there always exists an almost optimal $\{x, y\}$ -path-decomposition satisfying specific properties (first, *contiguous*, then *g-contiguous* (Section 4.2.2) and finally *LtR g-contiguous* (Section 4.2.3), see formal definitions below). Then, a dynamic programming algorithm to compute such an optimal $\{x, y\}$ -path-decomposition is presented (Section 4.2.4). We first need further notation presented in Section 4.2.1.

4.2.1 Notation

Let $x = \{x_1, x_2\}$ and $y = \{y_1, y_2\}$ (to simplify the presentation, we assume that $x, y \in E_{out}$. If $x \in V$ (resp., $y \in V$), it is sufficient to set $x_1 = x_2$ (resp., $y_1 = y_2$)). Let F (the internal face containing x and y) consists of two internally disjoint paths P_{up} between x_1 and y_1 and P_{down} between x_2 and y_2 (if x and y are edges, they may share one vertex, in which case, we assume that $x_2 = y_2$ and P_{down} is reduced to x_2).

Let \mathcal{C} be the set of connected components of $G \setminus F$. Let \mathcal{C}_{up} (resp., \mathcal{C}_{down}) be the set of connected components C of $G \setminus F$ such that $N(C) \subseteq V(P_{up})$ (resp., $N(C) \subseteq V(P_{down})$). For every $C \in \mathcal{C}_{up} \cup \mathcal{C}_{down}$, let $\bar{C} = C \cup N(C)$ and let $s_C = N(C)$ (abusing the notation, s_C will be either an edge or a vertex of F).

Lemma 11 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G .*

If $pl(G, x, y) \leq k$, then there exists an $\{x, y\}$ -path-decomposition $D' = (X_1, \dots, X_p)$ of G with length at most k such that, for every $C \in \mathcal{C}$, if $X_i \cap C \neq \emptyset$, then $s_C \in X_i$ (or $s_C \subseteq X_i$ if s_C is an edge).

Proof. Consider an $\{x, y\}$ -path-decomposition $D' = (X_1, \dots, X_p)$ of G with length at most k that maximizes the number of components of $C \in \mathcal{C}$ that satisfy the property that if $X_i \cap C \neq \emptyset$, then $s_C \in X_i$.

For purpose of contradiction, let us assume that there exists $C \in \mathcal{C}$ that does not satisfy this property. W.l.o.g., assume that $C \in \mathcal{C}_{up}$. Let $1 \leq i \leq j \leq p$ be the smallest (largest respectively) integer such that $C \cap X_i \neq \emptyset$ ($C \cap X_j \neq \emptyset$, resp.). Note that, for every $i \leq h \leq j$, $X_h \cap V(P_{down}) \neq \emptyset$, and, moreover, for every vertices $u \in s_C$, $v \in C$ and $w \in V(P_{down})$, $dist(u, w) \leq dist(v, w)$ and $d(v, u) \leq d(v, w)$.

Then, $(X_1, \dots, X_{i-1}, X_i \cup s_C, \dots, X_j \cup s_C, X_{j+1}, \dots, X_p)$ is an $\{x, y\}$ -path-decomposition $D' = (X_1, \dots, X_p)$ of G with length at most k , contradicting the maximality of D' . \blacksquare

Let C be a component of $G \setminus F$.

- If s_C is a vertex, let $d_C = \max_{v \in C \cup s_C} dist(v, s_C)$ and let h_C^* be any vertex such that $dist(h_C^*, s_C) = d_C$.

- If s_C is an edge, let $s_C = \{l_C, r_C\}$ and let $d_C = \max_{v \in C \cup s_C} \max\{\text{dist}(v, r_C), \text{dist}(v, l_C)\}$ and let $\mathcal{M}_C = \{v \in C \cup s_C \mid \max\{\text{dist}(v, r_C), \text{dist}(v, l_C)\} = d_C\}$. Note that, for every $v \in C \cup s_C$, $\text{dist}(v, l_C) - 1 \leq \text{dist}(v, r_C) \leq \text{dist}(v, l_C) + 1$. If there exists $v \in \mathcal{M}_C$ such that $\text{dist}(v, r_C) = \text{dist}(v, l_C) = d_C$, then let h_C^* be such a vertex. Otherwise, let h_C^* be any vertex of \mathcal{M}_C .

If s_C is a vertex or if there exists a vertex $v \in C \cup s_C$ with $\text{dist}(v, r_C) = \text{dist}(v, l_C) = d_C$, we say that C is a *convenient* component.

Claim 6 *Let C be a connected component of $G \setminus F$ and let $v \in C$ and $u \in G \setminus C$. If C is convenient or $v \notin \mathcal{M}_C$, $\text{dist}(v, u) \leq \text{dist}(h_C^*, u)$. Otherwise, $\text{dist}(v, u) \leq \text{dist}(h_C^*, u) + 1$.*

4.2.2 Toward g -contiguous decompositions

Let $D = (X_1, \dots, X_p)$ be any $\{x, y\}$ -path-decomposition of G and let $C \in \mathcal{C}$. The component C is said *contiguous* (with respect to D) if there exist $1 \leq a_C \leq b_C \leq p$ such that (1) $C \cap X_i = \emptyset$ if and only if $i \notin \{a_C, \dots, b_C\}$, and $s_C \in X_j$ for all $a_C \leq j \leq b_C$, and (2) there exists $R_C \subseteq V(F)$ such that $X_i \setminus C = R_C$ for every $a_C \leq i \leq b_C$. Intuitively, C is contiguous w.r.t. D if, once a vertex of C has been introduced in D , no vertex of $G \setminus C$ can be introduced in D before all vertices of C have been introduced.

A path-decomposition D is *contiguous* if every component of $G \setminus F$ is contiguous w.r.t. D .

In what follows, we show that there always exists an optimal (or almost optimal) $\{x, y\}$ -path-decomposition of G which is contiguous. Let us start with the following constrained (we assume that all components of $G \setminus F$ are convenient) but more favorable case (and easier to prove).

Theorem 4 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{\text{out}} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G . Moreover, let us assume that all components of $G \setminus F$ are convenient.*

If $pl(G, x, y) \leq k$, then there exists a contiguous $\{x, y\}$ -path-decomposition D' of G with length at most k .

Proof. Given any $\{x, y\}$ -path-decomposition $D = (X_1, \dots, X_p)$ of G with length at most k , let $\mathcal{Q}(D) \subseteq \mathcal{C}$ be a set of components C of \mathcal{C} such that there exist $1 \leq a_C \leq b_C \leq p$ such that:

1. $C \cap X_i \neq \emptyset$ if and only if $a_C \leq i \leq b_C$, and $s_C \in X_j$ for all $a_C \leq j \leq b_C$, and
2. there exists $R_C \subseteq F \cup \bigcup_{C' \notin \mathcal{Q}(D)} C'$ such that, for every $a_C \leq i \leq b_C$, $X_i \setminus C = R_C$.

Let D be an $\{x, y\}$ -path-decomposition of G with length k . Then, such a set $\mathcal{Q}(D)$ is well defined (possibly, $\mathcal{Q}(D)$ is empty).

Let us consider an $\{x, y\}$ -path-decomposition $D' = (X_1, \dots, X_p)$ of G with length at most k , and that maximizes $|\mathcal{Q}(D')|$. If $\mathcal{Q}(D') = \mathcal{C}$, then D' is the desired path-decomposition.

For purpose of contradiction, let us assume that $\mathcal{C} \setminus \mathcal{Q}(D') \neq \emptyset$. Let $C \in \mathcal{C} \setminus \mathcal{Q}(D')$ and let $1 \leq i \leq p$ be the smallest integer such that $h_C^* \in X_i$. Note that, by Lemma 11, $s_C \in X_i$. Note also that, there is no $C' \in \mathcal{Q}(D')$ such that $a_{C'} < i \leq b_{C'}$ by the second property above.

Let $Y = (D' \cap C) \cup (s_C \cup (X_{i-1} \cap X_i) \setminus C)$. by Claim 6, and the fact that C is convenient, $\ell(Y) \leq k$.

Therefore, $D'' = (X_1 \setminus C, \dots, X_{i-1} \setminus C) \odot Y \odot (X_i \setminus C, \dots, X_p \setminus C)$ is an $\{x, y\}$ -path-decomposition of G , with length at most k , and such that $\mathcal{Q}(D') \cup \{C\} \subseteq \mathcal{Q}(D'')$. This contradicts the maximality of $|\mathcal{Q}(D')|$. \blacksquare

Next, we include the cases when not all components of $G \setminus F$ are convenient. We show that it may imply an increase of +1 of the length of the path-decompositions. Later, we show that this increase cannot be avoided.

Theorem 5 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G .*

If $pl(G, x, y) \leq k$, then there exists a contiguous $\{x, y\}$ -path-decomposition D' of G with length at most $k + 1$.

Proof. Let $D = (X_1, \dots, X_p)$ be any $\{x, y\}$ -path-decomposition of G with length at most $k + 1$, such that, for every $1 \leq i \leq p$, every $C \in \mathcal{C}$ and every $u, v \in C \cap X_i$, we have that $dist(u, v) \leq k$ (Property (*)).

Let $\mathcal{Q}(D) \subseteq \mathcal{C}$ be a set of components C of \mathcal{C} such that there exist $1 \leq a_C \leq b_C \leq p$ such that:

1. $C \cap X_i \neq \emptyset$ if and only if $a_C \leq i \leq b_C$, and $s_C \in X_j$ for all $a_C \leq j \leq b_C$, and
2. there exists $R \subseteq F \cup \bigcup_{C' \notin \mathcal{Q}(D)} C'$ such that, for every $a_C \leq i \leq b_C$, $X_i \setminus C = R_C$, and
3. for every $1 \leq j \leq p$ such that there exists no $C \in \mathcal{Q}(D)$ with $a_C \leq j \leq b_C$, then $\ell(X_j) \leq k$.
Moreover, if there exist $C, C' \in \mathcal{Q}(D)$ with $b_{C'} = j = a_C - 1$, then $\ell(X_j \cap X_{j+1}) \leq k$.

Let D be an $\{x, y\}$ -path-decomposition of G with length k . In particular, note that D trivially satisfies Property (*). Then, such a set $\mathcal{Q}(D)$ is well defined (possibly, $\mathcal{Q}(D)$ is empty).

Let us consider an $\{x, y\}$ -path-decomposition $D' = (X_1, \dots, X_p)$ of G with length at most $k + 1$ and satisfying Property (*), and that maximizes $|\mathcal{Q}(D')|$. If $\mathcal{Q}(D') = \mathcal{C}$, then D' is the desired path-decomposition.

For purpose of contradiction, let us assume that $\mathcal{C} \setminus \mathcal{Q}(D') \neq \emptyset$. Let $C \in \mathcal{C} \setminus \mathcal{Q}(D')$ and let $1 \leq i \leq p$ be the smallest integer such that $h_C^* \in X_i$. Note that, by Lemma 11, $s_C \in X_i$. Note also that, there is no $C' \in \mathcal{Q}(D')$ such that $a_{C'} < i \leq b_{C'}$ by the second property above.

Let $Y = (D' \cap C) \cup (s_C \cup (X_{i-1} \cap X_i) \setminus C)$. By Property (*) and Claim 6 and third item above, $\ell(Y) \leq k + 1$ (if C is convenient, we even have $\ell(Y) \leq k$).

Therefore, $D'' = (X_1 \setminus C, \dots, X_{i-1} \setminus C) \odot Y \odot (X_i \setminus C, \dots, X_p \setminus C)$ is an $\{x, y\}$ -path-decomposition of G , with length at most $k + 1$ and satisfying Property (*), and such that $\mathcal{Q}(D') \cup \{C\} \subseteq \mathcal{Q}(D'')$. This contradicts the maximality of $|\mathcal{Q}(D')|$. \blacksquare

Unfortunately, previous theorem cannot be improved since there are 2-connected outerplanar graphs G and $x, y \in E_{out}$, such that every contiguous $\{x, y\}$ -path-decomposition has length at least $pl(G, x, y) + 1$.

Lemma 12 *There exists 2-connected outerplanar graphs G and $x, y \in E_{out}$ such that every contiguous $\{x, y\}$ -path-decomposition of G has length at least $pl(G, x, y) + 1$.*

Proof. Let G be the graph depict in Figure 2. Let C be the unique component of $G \setminus F$. Let us denote the vertices of s_C by l_C and r_C as depicted in Figure 2. Let us first describe a $\{x, y\}$ -path-decomposition D of G with length 10. Let $X_1 = \{x\}$, $X_2 = C_l \cup C_m \cup L$ where L is the (x_2, r_c) -path in F not containing y_1 and y_2 . Let $X_3 = C_m \cup \text{down}$ where down is the (x_2, y_2) -path in F not containing x_1 and y_1 . Let $X_4 = C_m \cup C_r \cup R$ where R is the (y_2, l_c) -path in F not containing x_1 and x_2 . Finally, let $X_5 = \{y\}$. Note that $D = (X_1, \dots, X_5)$ is a $\{x, y\}$ -path-decomposition of length 10. Moreover, D is not contiguous since for X_2 and X_4 contains both vertices from C but $X_2 \setminus V(C) \neq X_4 \setminus V(C)$.

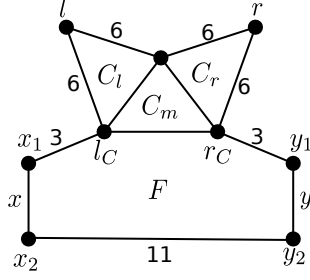


Figure 2: Example of a graph G and $x, y \in E_{out}$ such that every contiguous $\{x, y\}$ -path-decomposition of G has length at least $pl(G, x, y) + 1$. Weight on edges represent the length of a path between the two endpoints of the edges.

Let us apply lemma 5 to D . Note that the contiguous $\{x, y\}$ -path-decomposition D' obtained, has every vertex from $V(C)$ either contained with L or with R . Hence, either l and y_2 are both in a bag, or r and x_2 are both in a bag. In every case, there is a bag containing two vertices at distance 11. ■

A contiguous $\{x, y\}$ -path-decomposition $D = (X_1, \dots, X_p)$ of G is said *g-contiguous* if, for every $C \in \mathcal{C}$, $(D_{a_C} \cap (\bar{C}), \dots, D_{b_C} \cap (\bar{C}))$ is an optimal greedy path-decomposition of $G[\bar{C}] = G[C \cup s_C]$ based on s_C (see Section 4.1.3).

Theorem 6 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G .*

If $pl(G, x, y) \leq k$, then there exists a g-contiguous $\{x, y\}$ -path-decomposition $D = (X_1, \dots, X_p)$ of G with length at most $k + 1$. Moreover, for every $C \in \mathcal{C}$ and $a_C \leq i \leq b_C$, $\ell(X_i) = k + 1$ only if $h_C^ \in X_i$ for some $h_C^* \in M_C$.*

Proof. By Theorem 5, there exists a contiguous $\{x, y\}$ -path-decomposition $D' = (X_1, \dots, X_p)$ of G with length at most $k + 1$.

Iteratively, for every component $C \in \mathcal{C}$, replace D' with $(X_1, \dots, X_{a_C-1}) \odot \mathcal{G}(C) \cup (X_{a_C-1} \cap X_{b_C+1}) \odot (X_{b_C+1}, \dots, X_p)$ where $\mathcal{G}(C)$ is an optimal greedy path-decomposition of $G[\bar{C}] = G[C \cup s_C]$ based on s_C . This remains a contiguous $\{x, y\}$ -path-decomposition of G since, for every $a_C \leq i \leq b_C$, $s_C \in X_i$ and $X_i \setminus C \subseteq V(F)$.

Moreover, by construction of D' (see the proof of Theorem 5), two vertices u and v in a bag X_i ($a_C \leq i \leq b_C$ for some $C \in \mathcal{C}$) may be at distance $k + 1$ only if $u \in C$ and $v \in V(F)$. Therefore, the result holds (including the last statement) by definition of greedy path-decomposition and by Claim 6. ■

4.2.3 Toward left-to-right g-contiguous decompositions

Recall that we are considering a n -node connected simple outerplanar graph $G = (V, E)$ and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G .

Let $x = \{x_1, x_2\}$ if x is an edge (or $x = x_1 = x_2$ if x is a vertex) and let $y_1, y_2 \in V$ be defined similarly for y . Let $P_{up} = (x_1 = u_1, \dots, u_t = y_1)$ and $P_{down} = (x_2 = d_1, \dots, d_s = y_2)$ be the two internally disjoint paths that consist of all vertices of the face F . Note that, it may be possible that $x_1 = y_1$ or $x_2 = y_2$ but not both (in what follows, we assume that $x_1 \neq y_1$).

For every $C \in \mathcal{C}_{up}$ (i.e., such that $s_C \in V(P_{up})$ or $s_C \subseteq V(P_{up})$), let l_C (resp., r_C) be the vertex of s_C that is closest (resp., furthest) to x_1 in P_{up} (or $l_C = r_C = s_C$ if s_C is a vertex). Similarly, for every $C \in \mathcal{C}_{down}$ (i.e., such that $s_C \in V(P_{down})$ or $s_C \subseteq V(P_{down})$), let l_C (resp., r_C) be the vertex of s_C that is closest (resp., furthest) to x_2 in P_{down} (or $l_C = r_C = s_C$ if s_C is a vertex).

An edge $e \in E(F) \setminus \{x, y\}$ is said *trivial* if there does not exist $C \in \mathcal{C}$ such that $e = s_C$ (note that an end of e may be a cut vertex, but there is no $C \in \mathcal{C}$ with $N(C) = e$). While trivial edges are not related to any component of $G \setminus F$, we need to include them in the analysis that follows. To unify the notation, let $\bar{\mathcal{C}} = \{\bar{C} = C \cup s_C \mid C \in \mathcal{C}\} \cup \{e \in E(F) \setminus \{x, y\} \mid e \text{ is a trivial edge}\}$. Intuitively, every trivial edge $e \in E(F) \setminus \{x, y\}$ may be seen as $e = s_C$ for some dummy empty component $C = \emptyset$. Similarly, let $\bar{\mathcal{C}}_{up} = \{\bar{C} = C \cup s_C \mid C \in \mathcal{C}_{up}\} \cup \{e \in E(P_{up}) \mid e \text{ is a trivial edge}\}$ and $\bar{\mathcal{C}}_{down} = \{\bar{C} = C \cup s_C \mid C \in \mathcal{C}_{down}\} \cup \{e \in E(P_{down}) \mid e \text{ is a trivial edge}\}$. Note that $\bar{\mathcal{C}} = \bar{\mathcal{C}}_{up} \cup \bar{\mathcal{C}}_{down}$.

Let $\mathcal{O}_{up} = (C_1^u, \dots, C_{s'}^u)$ be any ordering of $\bar{\mathcal{C}}_{up}$ such that, if $l_{C_i^u}$ is strictly closer to x_1 in P_{up} than $l_{C_j^u}$ and/or if $r_{C_i^u}$ is strictly closer to x_1 in P_{up} than $r_{C_j^u}$, then $i < j$. Similarly, let $\mathcal{O}_{down} = (C_1^d, \dots, C_{s'}^d)$ be any ordering of $\bar{\mathcal{C}}_{down}$ such that, if $l_{C_i^d}$ is strictly closer to x_2 in P_{down} than $l_{C_j^d}$ and/or if $r_{C_i^d}$ is strictly closer to x_2 in P_{down} than $r_{C_j^d}$, then $i < j$. Intuitively, we order the components of \mathcal{C}_{up} and the trivial edges of P_{up} from x_1 to y_1 (resp., of \mathcal{C}_{down} and the trivial edges of P_{down} from x_2 to y_2), from “left to right” (“from x to y ”). Note that for every two components $C, C' \in \mathcal{C}$ such that $s_C = s_{C'} \in V$, the relative order between C and C' in \mathcal{O}_{up} (resp., in \mathcal{O}_{down} if $s_C \in V(P_{down})$) is not relevant (but all connected components C' with $s_C = s_{C'}$ are consecutive in \mathcal{O}_{up} (resp., in \mathcal{O}_{down})).

In this section, we only consider g -contiguous $\{x, y\}$ -path-decompositions $D = (X_1, \dots, X_p)$ of G . That is, for every $C \in \mathcal{C}$, there exist $1 \leq a_C \leq b_C \leq p$, and an interval $I_C = [a_C, b_C]$ such that $X_i \cap C \neq \emptyset$ if and only if $i \in I_C$, $s_C \in X_i$ for all $i \in I_C$ and $X_i \setminus C \subseteq F$ for all $i \in I_C$. In particular, $I_C \cap I_{C'} = \emptyset$ for all distinct $C, C' \in \mathcal{C}$. We say that C appears in D in the bag X_{a_C} . Moreover, $(X_{a_C} \cap \bar{C}, \dots, (X_{b_C} \cap \bar{C}))$ is a greedy path-decomposition of $G[\bar{C}]$ based on s_C . Recall also that we may assume that the property of the last statement in Theorem 6 holds.

By definition, D induces a total order $\mathcal{O}_D = (\bar{C}_1, \dots, \bar{C}_b)$ on $\bar{\mathcal{C}}$ such that, for any $1 \leq i < j \leq b$, \bar{C}_i appears in D before \bar{C}_j (i.e., $b_{C_i} < a_{C_j}$). We aim at considering such g -contiguous $\{x, y\}$ -path-decompositions D such that the total orders \mathcal{O}_D they induce satisfy some extra property defined below.

Let $H = H^1 \cup H^2$ be a ground set with $H^1 \cap H^2 = \emptyset$. Let $\mathcal{O} = (H_1, \dots, H_q)$ be a total ordering on H , and let \mathcal{O}^i be a total ordering of H^i for $i \in \{1, 2\}$. A prefix $\mathcal{P} = (H_1, \dots, H_{q'})$ ($q' \leq q$) of H is *compatible* with \mathcal{O}^i if $\mathcal{P} \cap H^i$ is a prefix of \mathcal{O}^i for $i \in \{1, 2\}$. If $q' = q$, then \mathcal{O} is said *compatible* with \mathcal{O}^1 and \mathcal{O}^2 .

A contiguous $\{x, y\}$ -path-decompositions D of G is said *LtR* (*left to right*) if \mathcal{O}_D is compatible with \mathcal{O}_{up} and \mathcal{O}_{down} . Another equivalent way to define *LtR* is as follows:

Claim 7 *A contiguous $\{x, y\}$ -path-decompositions $D = (X_1, \dots, X_p)$ of G is LtR if and only if (1), for every $C_i^u, C_j^u \in \mathcal{O}_{up}$ (resp., $C_i^d, C_j^d \in \mathcal{O}_{down}$) with $i < j$, $b_{C_i^u} < a_{C_j^u}$ (resp., with $b_{C_i^d} < a_{C_j^d}$) and, moreover, (2), for every $C \in \bar{\mathcal{C}}_{up}$ (resp., $\bar{\mathcal{C}}_{down}$), and $i \in I_C$, $X_i \cap F = \{l_C, r_C, f_C\}$ where f_C is one vertex of $V(P_{down})$ (resp., of $V(P_{up})$).*

In what follows, we will iteratively transform a given g -contiguous $\{x, y\}$ -path-decomposition of G . While, during these transformations, the obtained path-decomposition will always remain a g -contiguous $\{x, y\}$ -path-decomposition of G , its length may be increased temporary. To deal with this difficulty, let us define the *weak length*, denoted by $wl(D)$ of a path-decomposition $D = (X_1, \dots, X_p)$. The *weak length*, denoted by $wl(X_i)$, of a bag X_i ($1 \leq i \leq p$) is

$\max_{u \in X_i, v \in X_i \cap Y} \text{dist}(u, v)$ where $Y = V(P_{up})$ (resp., $Y = V(P_{down})$) if $a_C \leq i \leq b_C$ for a component $\bar{C} \in \bar{\mathcal{C}}_{down}$ (resp., $C \in \bar{\mathcal{C}}_{up}$). Then, $wl(D) = \max_{i \leq p} wl(X_i)$.

Lemma 13 *Let D be a LtR g -contiguous $\{x, y\}$ -path-decomposition of G , satisfying the last statement of Theorem 6, and of weak length k . Then, $\ell(D) \leq k$.*

Proof. This holds by definition of LtR and by the last statement of Theorem 6. \blacksquare

Next theorem roughly says that, from a g -contiguous $\{x, y\}$ -path-decomposition, we can add the property that it is LtR without increasing the length.

Theorem 7 *Let $G = (V, E)$ be a connected simple outerplanar graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G .*

Let us assume that there exists a g -contiguous $\{x, y\}$ -path-decomposition $D = (X_1, \dots, X_p)$ of G with length k and such that, for every $C \in \mathcal{C}$ and $a_C \leq i \leq b_C$, $\ell(X_i) = k$ only if $h_C^ \in X_i$ for some $h_C^* \in \mathcal{M}_C$.*

Then, there exists a LtR g -contiguous $\{x, y\}$ -path-decomposition of G with length at most k .

Proof. Let $D = (X_1, \dots, X_p)$ be a g -contiguous $\{x, y\}$ -path-decomposition of G with weak length k . We say that D satisfies Property (*) if, for every $C \in \mathcal{C}$ and $a_C \leq i \leq b_C$, $wl(X_i) = k$ only if $h_C^* \in X_i$ for some $h_C^* \in \mathcal{M}_C$.

Recall that \mathcal{O}_{up} (resp., \mathcal{O}_{down}) are not uniquely defined in the sense that the order between two components $C, C' \in \mathcal{C}_{up}$ (resp., $C, C' \in \mathcal{C}_{down}$) with $s_C = s_{C'} \in V$ is arbitrary (but all such components are consecutive in \mathcal{O}_{up} (resp., \mathcal{O}_{down})).

Let $D = (X_1, \dots, X_p)$ be g -contiguous $\{x, y\}$ -path-decomposition of G with weak length at most k satisfying Property (*) and let \mathcal{O}_{up} and \mathcal{O}_{down} be orderings of $\bar{\mathcal{C}}_{up}$ and $\bar{\mathcal{C}}_{down}$ (as defined previously) that maximize $1 \leq h \leq p$ such that (X_1, \dots, X_h) is compatible with \mathcal{O}_{up} and \mathcal{O}_{down} . Note that, if $h = p$, then D is the desired path-decomposition. Hence, for purpose of contradiction, let us assume that $h < p$.

Note that, because D is a contiguous $\{x, y\}$ -path-decomposition and because (X_1, \dots, X_h) is compatible with \mathcal{O}_{up} and \mathcal{O}_{down} , there exist $1 \leq i \leq s$ and $1 \leq j \leq t$ (recall that t and s are the number of vertices of P_{up} and P_{down} respectively) such that $X_h \cap X_{h+1} = \{u_i, d_j\}$.

Let $\mathcal{O}_D = \mathcal{O} \odot (C_1, \dots, C_q)$ where \mathcal{O} is the prefix of \mathcal{O}_D that corresponds to the components appearing in (X_1, \dots, X_h) . W.l.o.g., let us assume that $C_1 \in \bar{\mathcal{C}}_{up}$. Let $\mathcal{O}_{up} = \mathcal{O}' \odot (C'_1, \dots, C'_{q'})$ where $\mathcal{O}' = \mathcal{O} \cap \bar{\mathcal{C}}_{up}$. By maximality of h , $C_1 \neq C'_1$. More precisely, $C_1 = C'_z$ for some $1 < z \leq q'$. There are two cases to be considered.

- First, let us assume that, for every $1 \leq \alpha < z$ and for all $h_C^* \in \mathcal{M}_C$, $\text{dist}(h_{C'_\alpha}^*, d_j) \leq k$.
Let

$$D' = (X_1, \dots, X_h) \odot ((X_{a_{C'_1}}, \dots, X_{b_{C'_1}}) \cap \bar{C}'_1) \cup \{d_j\} \odot ((X_{a_{C'_2}}, \dots, X_{b_{C'_2}}) \cap \bar{C}'_2) \cup \{d_j\} \odot \dots \odot ((X_{a_{C'_{z-1}}}, \dots, X_{b_{C'_{z-1}}}) \cap \bar{C}'_{z-1}) \cup \{d_j\} \odot ((X_{h+1}, \dots, X_p) \setminus ((\bigcup_{1 \leq \alpha < z} \bar{C}'_\alpha) \setminus \{l_{C'_z}\})).$$

Intuitively, all components that are between (in \mathcal{O}_{up}) the last component of \mathcal{O} and C_1 are “moved” just before C_1 in the decomposition (in D all these components were appearing after C_1).

Because D is g -contiguous and satisfies Property (*), then D' is a g -contiguous $\{x, y\}$ -path-decomposition of G satisfying Property (*). Moreover, its weak length is at most k . In particular, for every bag B of $((X_{a_{C'_\alpha}}, \dots, X_{b_{C'_\alpha}}) \cap \bar{C}'_\alpha) \cup \{d_j\}$ for $1 \leq \alpha < z$, $wl(B) \leq k$ because D satisfies Property (*) and because, by assumption, $dist(h_{C'_\alpha}^*, d_j) \leq k$.

To conclude this case, D' is g -contiguous $\{x, y\}$ -path-decomposition of G with weak length at most k satisfying Property (*) and with a larger prefix than D that is compatible with \mathcal{O}_{up} and \mathcal{O}_{down} , contradicting the maximality of h .

- Else, for every decomposition D defined as above and maximizing h (where $z := z(D)$, defined as above, depends on D), there exists an integer $1 \leq \alpha < z(D)$ and a vertex $h_{C'_\alpha}^* \in \mathcal{M}_{C'_\alpha}$ such that $dist(h_{C'_\alpha}^*, d_j) > k$ (otherwise, we are back to the previous case). Let $\alpha(D)$ be the smallest such integer α for the decomposition D .

Let $1 < \alpha^*(D) \leq q$ be such that $C'_{\alpha(D)} = C_{\alpha^*(D)}$.

Consider such a decomposition D (still maximizing h) that minimizes $\alpha^*(D)$. From now on, we denote the integer $\alpha(D)$ (for this particular decomposition D) by α and $\alpha^*(D)$ is denoted by α^* .

Let $\alpha < \beta \leq z \leq \gamma \leq q'$ be defined such that $[\beta, \gamma]$ is the inclusion-maximal interval (containing z) such that every component C'_m with $m \in [\beta, \gamma]$ appears before C_{α^*} in \mathcal{O}_D (i.e., for every $m \in [\beta, \gamma]$, setting m' such that $C'_m = C_{m'}$, then $1 \leq m' < \alpha^*$). Let $C_{l_{C'_\beta}}$ denote the set of component C from \mathcal{C} such that $l_{C'_\beta} \in s_C$, $C \notin I$ and $C \leq_D \alpha^*$, i.e. the component C that appears before C_{α^*} in D , which will not be moved, and that contains $l_{C'_\beta}$ (note that we cannot remove $l_{C'_\beta}$ from the bags of the component in $C_{l_{C'_\beta}}$ without breaking a property of path-decomposition). Note that C_{α^*} can be in $C_{l_{C'_\beta}}$. Let $C_{r_{C'_\gamma}}$ denote the set of component C from \mathcal{C} such that $r_{C'_\gamma} \in s_C$, $C \notin I$ and $C \leq_D \alpha^*$.

- First, let us assume that, $C_{l_{C'_\beta}} = C_{r_{C'_\gamma}} = \emptyset$. Let

$$D' = (X_1, \dots, X_h) \odot ((X_{h+1}, \dots, X_{a_{C_{\alpha^*}}-1}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \odot (X_{a_{C_{\alpha^*}}}, \dots, X_{b_{C_{\alpha^*}}}) \odot$$

$$((X_{h+1}, \dots, X_{a_{C_{\alpha^*}}-1}) \cap \bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m) \cup (X_{b_{C_{\alpha^*}}} \cap V(P_{down})) \odot (X_{b_{C_{\alpha^*}}+1}, \dots, X_p).$$

Intuitively, all components $C'_\beta, \dots, C'_\gamma$ (and in particular, $C_1 = C'_z$) that were appearing before C_{α^*} in D (but that are greater than C_{α^*} in \mathcal{O}_{up}) are “moved” after C_{α^*} in D .

Because D is g -contiguous and satisfies Property (*), then D' is a g -contiguous $\{x, y\}$ -path-decomposition of G satisfying Property (*). In particular, each edge of G belongs to some bag because we have ensured that all components intersecting $\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m$ do not appear in $(X_{h+1}, \dots, X_{a_{C_{\alpha^*}}-1})$. It remains to prove that its weak length is at most k .

We will prove that every bag in $((X_{h+1}, \dots, X_q) \cap \bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m) \cup (X_{b_{C_{\alpha^*}}} \cap V(P_{down}))$ has weak length at most k (that are the only bags where some vertices may be added compared with the bags of D). Since we are considering the weak length, we actually need to prove that, for every $v \in \bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m$ and every $w \in X_{b_{C_{\alpha^*}}} \cap V(P_{down})$,

$dist(v, w) \leq k$. Actually, we will show that $dist(v, w) \leq dist(h_{C'_\alpha}^*, w)$ (note that $dist(h_{C'_\alpha}^*, w) \leq k$ since w belongs to every bag in $X_{a_{C'_\alpha}}, \dots, X_{b_{C'_\alpha}}$).

Note that, because D is a path-decomposition, $X_h \cap X_{h+1} \cap V(P_{down}) = \{d_j\}$ and $y_2 \in X_p$, then w is between d_j and y_2 in P_{down} . Moreover, since $dist(h_{C'_\alpha}^*, d_j) > k$ and $dist(h_{C'_\alpha}^*, w) \leq k$, then the shortest path between w and $h_{C'_\alpha}^*$ goes through y_2 .

Let $\beta \leq m \leq \gamma$ such that $v \in \bar{C}'_m$ and let $1 \leq \delta < a_{C'_\alpha}$ be such that $v \in X_\delta$ (in D). Because D is a path-decomposition and $d_j \in X_h$ and $w \in X_{a_{C'_\alpha}}$, there must be a vertex $w' \in X_\delta$ which is between d_j and w in P_{down} , and so $dist(v, w') \leq k$ since D has weak length at most k . If the shortest path between v and w' goes through y_2 , we get that $dist(v, w) \leq dist(v, w') \leq k$ and we are done. Otherwise, w is strictly between w' and y_2 (in particular $w \neq w'$) (because the shortest path between $h_{C'_\alpha}^*$ and w goes through y_2 , the one between v and w' goes through x_2 and $s_{C'_\alpha}$ is closer to x_2 than $s_{C'_m}$). Note also that w' belongs to every bag in $X_{a_{C'_m}}, \dots, X_{b_{C'_m}}$ and, in particular, one of this bag contains a vertex $h_{C'_m}^*$, and so $dist(h_{C'_m}^*, w') \leq k$.

To sum up, $dist(w', l_{C'_m}) + dist(l_{C'_m}, h_{C'_m}^*) = dist(h_{C'_m}^*, w') \leq k < dist(h_{C'_\alpha}^*, d_j) = \leq dist(w', l_{C'_\alpha}) + dist(l_{C'_\alpha}, h_{C'_\alpha}^*)$. Because $l_{C'_\alpha}$ is between x_2 and $l_{C'_m}$ in P_{up} and the shortest path between $l_{C'_m}$ and w' goes through x_2 , we get that $dist(w', l_{C'_m}) \geq dist(w', l_{C'_\alpha})$ and so $dist(l_{C'_m}, h_{C'_m}^*) < dist(l_{C'_\alpha}, h_{C'_\alpha}^*)$. Finally, $k \geq dist(h_{C'_\alpha}^*, w) = dist(h_{C'_\alpha}^*, r_{C'_\alpha}) + dist(r_{C'_\alpha}, w) \geq dist(r_{C'_m}, h_{C'_m}^*) + dist(r_{C'_m}, w) = dist(h_{C'_m}^*, w)$. The last inequality comes from the fact that $r_{C'_m}$ is between y_1 and $r_{C'_\alpha}$ in P_{up} and the shortest path between $l_{C'_\alpha}$ and w goes through y_1 and y_2 . Hence, we get that $dist(h_{C'_m}^*, w) \leq k$ and so, $dist(v, w) \leq k$ by Property (*).

It remains to show that D' contradicts the minimality of α^* . Let C be the component that appears in D' just after X_h .

* First, let us assume that $C \in \mathcal{C}_{up}$.

If $C \leq C'_\alpha$ in \mathcal{O}_{up} , then by definition of α , D' corresponds to the first case of the proof of the Theorem, i.e., C and all components smaller than C in \mathcal{O}_{up} can be moved just after X_h in D' (recall that all these components can be added with d_j by definition of α).

Otherwise, by definition of α (and of d_j), we have that $\alpha(D') = \alpha(D) = \alpha$ and, then $C_{\alpha^*(D')} = C_{\alpha^*(D)}$. Since $C_{\alpha^*(D)}$ is “closer” from X_h in D' than in D (since D' is obtained from D by moving at least C_1 after $C_{\alpha^*(D)}$), we get that $\alpha^*(D') < \alpha^*(D)$, contradicting the minimality of $\alpha^*(D)$.

* If $C \in \mathcal{C}_{down}$, then we repeat the process. Either we fall in the first case, which contradicts the maximality of h , or we have to repeat the transformation of the second case. This leads to a new decomposition D'' with same prefix (X_1, \dots, X_h) and a component C' that appears just after this prefix in D'' . If $C' \in \mathcal{C}_{down}$, then applying the above paragraph with D' and D'' instead of D and D' leads to a contradiction. Otherwise, since the prefix (and so d_j) is the same, then applying the paragraph above for C' also contradicts the minimality of $\alpha^*(D)$ (i.e., $\alpha^*(D'') < \alpha^*(D)$).

– Second, let us assume that $r_{C'_\alpha} = l_{C'_\beta}$ and that there exists a component $C \notin \{C'_\beta, \dots, C'_\gamma\}$ such that $r_C = l_{C'_\beta}$ that appears before C'_α in D (because $C \notin \{C'_\beta, \dots, C'_\gamma\}$, this implies that $C < C'_\alpha$ in \mathcal{O}_{up}). Let $1 \leq \delta < \alpha$ be the smallest integer such that C'_δ is such a component C , and let $h + 1 \leq \delta' < a_{C'_\alpha}$ be such that $\delta' = a_{C'_\delta}$. Then, let

$$\begin{aligned}
D' &= (X_1, \dots, X_h) \odot ((X_{h+1}, \dots, X_{b_{C'_\delta}}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \odot \\
&(((X_{b_{C'_\delta}+1}, \dots, X_{a_{C_{\alpha^*}-1}}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \cup \{l_{C'_\beta}\}) \odot (X_{a_{C_{\alpha^*}}}, \dots, X_{b_{C_{\alpha^*}}}) \odot \\
&((X_{h+1} \dots, X_{a_{C_{\alpha^*}-1}} \cap \bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m) \cup (X_{b_{C_{\alpha^*}}} \cap V(P_{down}))) \odot (X_{b_{C_{\alpha^*}+1}}, \dots, X_p).
\end{aligned}$$

With similar arguments as in the previous case, D' contradicts the minimality of α .

- Second, let us assume that $C_{l_{C'_\beta}} \neq \emptyset$, and that $C_{r_{C'_\beta}} = \emptyset$ (note that the case when $C_{l_{C'_\beta}} = \emptyset$, and $C_{r_{C'_\beta}} \neq \emptyset$ is symmetric). Let us denote by C'_γ the first component in D that is also in $C_{l_{C'_\beta}}$. Then, let

$$\begin{aligned}
D' &= (X_1, \dots, X_h) \odot ((X_{h+1}, \dots, X_{b_{C'_\delta}}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \odot \\
&(((X_{b_{C'_\delta}+1}, \dots, X_{a_{C_{\alpha^*}-1}}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \cup \{l_{C'_\beta}\}) \odot (X_{a_{C_{\alpha^*}}}, \dots, X_{b_{C_{\alpha^*}}}) \odot \\
&((X_{h+1} \dots, X_{a_{C_{\alpha^*}-1}} \cap \bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m) \cup (X_{b_{C_{\alpha^*}}} \cap V(P_{down}))) \odot (X_{b_{C_{\alpha^*}+1}}, \dots, X_p).
\end{aligned}$$

With similar arguments as in the previous case, D' contradicts the minimality of α .

- Third, let us assume that $C_{l_{C'_\beta}} \neq \emptyset$, and that $C_{r_{C'_\beta}} \neq \emptyset$. Let us denote by C^l_γ the first component in D that is also in $C_{l_{C'_\beta}}$. Let us denote by C^r_γ the first component in D that is also in $C_{l_{C'_\beta}}$. W.l.o.g., let us assume that C^l_γ appears before C^r_γ in D .

$$\begin{aligned}
D' &= (X_1, \dots, X_h) \odot ((X_{h+1}, \dots, X_{b_{C'_\delta}}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \odot \\
&(((X_{b_{C'_\delta}}, \dots, X_{b_{C'_\delta}-1}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \cup \{l_{C'_\beta}\}) \odot \\
&(((X_{b_{C'_\delta}}, \dots, X_{a_{C_{\alpha^*}-1}}) \setminus (\bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m)) \cup \{l_{C'_\beta}, r_{C'_\gamma}\}) \odot (X_{a_{C_{\alpha^*}}}, \dots, X_{b_{C_{\alpha^*}}}) \odot \\
&((X_{h+1} \dots, X_{a_{C_{\alpha^*}-1}} \cap \bigcup_{\beta \leq m \leq \gamma} \bar{C}'_m) \cup (X_{b_{C_{\alpha^*}}} \cap V(P_{down}))) \odot (X_{b_{C_{\alpha^*}+1}}, \dots, X_p).
\end{aligned}$$

With similar arguments as in the previous case, D' contradicts the minimality of α . ■

4.2.4 Compute optimal left-to-right g -contiguous decompositions in polynomial-time

Finally, we show that *LtR* g -contiguous $\{x, y\}$ -path-decompositions can be computed efficiently.

Theorem 8 *Let $G = (V, E)$ be a connected simple outerplanar n -node graph and $x, y \in E_{out} \cup V$ such that $G \setminus x$ and $G \setminus y$ are connected, $x \neq y$, $x \notin y$, $y \notin x$ and x and y lie on the same internal face F of G . Let us assume that $pl(G, x, y) \leq k$.*

*Then, an *LtR* g -contiguous $\{x, y\}$ -path-decomposition of G with length at most k can be computed in time $O(n + k^2)$.*

Proof. First, the connected components C (and $\bar{C} = C \cup s_C$) of $G \setminus F$ can be computed in linear-time. Note that these components are pairwise edge-disjoint. By Theorem 3, in global time $O(|E|)$, an optimal greedy path-decomposition D_C based on s_C can be computed for each such a component \bar{C} .

For each cut vertex $v \in V(F)$, let \mathcal{C}_v be the set of the connected components of $G \setminus v$ that do not contain x (nor y) and let $C_v \in \mathcal{C}_v$ (note that $v = s_{C_v}$) be a component that maximizes $dist(h_{C_v}^*, v)$. Let G' be obtained from G by removing, for every cut-vertex $v \in V(F)$, all components of \mathcal{C}_v but C_v . That is, for every cut-vertex v of F , there is a single component of $G' \setminus v$ not containing x nor y .

We first compute a path-decomposition for G' that we then extend to a path-decomposition of G .

Let $(\bar{C}_1^u, \dots, \bar{C}_q^u)$ be the components C such that $s_C \in P_{up}$ ordered from “left to right” (from x_1 to y_1 as in previous section), and let $(\bar{C}_1^d, \dots, \bar{C}_{q'}^d)$ be the components C such that $s_C \in P_{down}$ ordered from x_2 to y_2 (we use the same notation as in Section 4.2.3).

Note that F induces an isometric subgraph of G (and of G') and so, by Lemma 1 and Theorem 2, $|F| = O(k)$. Therefore, since every cut-vertex of G' corresponds to a single component, it follows that $q, q' = O(k)$.

For every $0 \leq i \leq q$, $0 \leq j \leq q'$, let $G[i, j]$ be the subgraph of G induced by $x = \{x_1, x_2\}$ and every component $\bar{C}_{i'}^u$, $1 \leq i' \leq i$ and every component $\bar{C}_{j'}^d$, $1 \leq j' \leq j$ and adding a new edge $e_{i,j}$ between $r_{\bar{C}_i^u}$ and $r_{\bar{C}_j^d}$ of length $dist_G(r_{\bar{C}_i^u}, r_{\bar{C}_j^d})$ (this artificial edge is used to keep the same distances as in G and G'). Let $D[i, j]$ denote an optimal *LtR* g -contiguous $\{x, e_{i,j}\}$ -path-decomposition of $G[i, j]$.

Let $D_1 = D[i-1, j] \odot D_{C_i^u} \cup \{r_{C_i^d}\}$ and let $D_2 = D[i, j-1] \odot D_{C_j^d} \cup \{r_{C_i^u}\}$. Let $D' \in \{D_1, D_2\}$ then, by definition of *LtR* g -contiguous path-decomposition, D' is an optimal *LtR* g -contiguous $\{x, e_{i,j}\}$ -path-decomposition of $G[i, j]$. Therefore, for every $0 \leq i \leq q$, $0 \leq j \leq q'$, $D[i, j]$ can be computed in constant time from $D[i-1, j]$ and $D[i, j-1]$. Indeed, computing the length of D_1 (resp., D_2) from the one of $D[i-1, j]$ (resp., of $D[i, j-1]$) only relies on the distance between $h_{C_j^d}^*$ and $r_{C_i^u}$ (resp., between $h_{C_i^u}^*$ and $r_{C_j^d}$) and these distances can be pre-computed in linear global time (for each component C , a single BFS from s_C is sufficient to determine h_C^* and its distance to s_C).

Hence, once the linear time pre-processings (computation of the greedy decompositions and of the distances) have done, an $\{x, y\}$ -path-decomposition $D[q, q']$ of G' with length at most k can be computed in time $O(k^2)$ by dynamic programming.

To conclude, to obtain the desired decomposition for G from $D[q, q']$, it is sufficient, for every cut-vertex $v \in V(F)$ and every component $C \in \mathcal{C}_v \setminus \{C_v\}$, to insert the greedy decomposition $D_C \cup x_v$ just after the one of D_{C_v} in $D[q, q']$ where $x_v \neq v$ is the (unique) vertex of F not in \bar{C}_v that appears in the same bags as \bar{C}_v in $D[q, q']$ (by maximality of $dist(h_{C_v}^*, v)$ when defining C_v , this does not increase the path-length). \blacksquare

4.3 Polynomial-time +1 approximation for computing the pathlength of outerplanar graphs

We are finally ready to prove our main theorem.

Theorem 9 *There exists an algorithm that, for every n -node connected outerplanar graph $G = (V, E)$, decides in time $O(n^3(n + k^2))$ whether $pl(G) > k$ or returns a path-decomposition of G with length at most $k + 1$.*

Proof. For every $x, y \in E_{out} \cup V$ (possibly $x = y$), the algorithm computes whether $pl(G, x, y) > k$ or computes an $\{x, y\}$ -path-decomposition of G length at most $k + 1$. By Lemma 3, such a decomposition with minimum length will be a path-decomposition of G with length at most $k + 1$. (We need to also consider the cases when x and y are vertices for the recursion below).

Let us fix $x, y \in E_{out} \cup V$ and let us assume that $pl(G, x, y) \leq k$, we present an algorithm that computes an $\{x, y\}$ -path-decomposition of G length at most $k + 1$ in time $O(n(n + k^2))$.

If $x = y$, the Algorithm *Greedy* (see Section 4.1.3) computes an optimal $\{x, y\}$ -path-decomposition of G in linear time by Theorem 3 and we are done. So let us assume that $x \neq y$.

First, in linear time, the cut-vertices and internal edges separating x and y are computed (this can be done, e.g., using SPQR trees [10]). Let $\{e_0 = x, e_1, \dots, e_{q-1}, e_q = y\}$ where $e_i \in E_{int} \cup V$ for every $0 < i < q$ be the set of those separators in order they are met when going from x to y . For every $0 \leq i < q$, $e_i \neq e_{i+1}$ (they may intersect) and either e_i and e_{i+1} share a same internal face F_i or, e_i and e_{i+1} are vertices and $\{e_i, e_{i+1}\} \in E$. Let \mathcal{C} be the set of connected components of $G \setminus e_i$ for some $0 \leq i \leq q$ that contain neither x nor y . That is, for every $C \in \mathcal{C}$, there exists $0 \leq i \leq q$ and $v \in e_i$ (or $v = e_i$ if e_i is a vertex) such that $N(C) = \{v\}$. Note that this decomposition into several connected components is done once for all at the beginning of the execution of the algorithm. In particular, it is not done anymore in the recursive calls described below and therefore it counts only for a linear time in the time complexity.

Assume first that $e_1 \neq y$. Let C' be the connected component of $G \setminus e_1$ containing (or intersecting if $e_1 \cap x \neq \emptyset$) x . Let $G_y = G[V \setminus C']$, let $\mathcal{C}_x \subseteq \mathcal{C}$ be the set of the connected components of $G \setminus x$ that do not contain y (they already have been computed above) and let $G_x = G[(C \cup e_1)]$ and let $G'_x = G_x \setminus \bigcup_{C \in \mathcal{C}_x} C$ be the subgraph induced by the vertices in $C \cup e_1$ that are not in some component of \mathcal{C}_x .

Our algorithm first recursively computes a $\{e_1, y\}$ -path-decomposition D_y of G_y with length at most $k + 1$ in time $O(|V(G_y)|(|V(G_y)| + k^2))$. Then, note that $G'_x \setminus x$ and $G'_x \setminus e_1$ are connected and $e_1 \neq x$ share a same face F_0 .

- If $x \in e_1$ or $e_1 \in x$, let D'_x be an optimal greedy $\{x, e_1\}$ -path-decomposition of G'_x in time $O(|E(G'_x)|)$ (see Section 4.1.3).
- Otherwise, the condition of Theorem 8 are fulfilled and an $\{x, e_1\}$ -path-decomposition D'_x of G'_x with length at most $k + 1$ can be computed in time $O(|V(G'_x)| + k^2)$.

Then, for every $C \in \mathcal{C}_x$ with $N(C) = \{v\}$, our algorithm computes an optimal greedy path-decomposition D_c of \bar{C} based on v . Using several times (one time per component of \mathcal{C}_x) some of the Lemmas 5-9 (depending on whether x and e_1 are edges or not, whether they intersect or not, and whether $N(C)$ intersects e_1 or not), an $\{x, e_1\}$ -path-decomposition D_x of G_x with length at most $k + 1$ can be computed in time $|\mathcal{C}_x|$ from D'_x and the decompositions D_c , $C \in \mathcal{C}_x$. Finally, from Lemma 10, the desired $\{x, y\}$ -path-decomposition of G with length at most $k + 1$

is obtained from D_x and D_y . So in total, in time $O(|V(G_y)|(|V(G_y)| + k^2)) + O(|V(G'_x)| + k^2) = O(n^3(n + k^2))$.

The last case is when $e_1 = y$. In that case, let G'_x be obtained from G by removing the component of $G \setminus x$ that do not contain y and also removing the components of $G \setminus y$ that do not contain x . An $\{x, y\}$ -path-decomposition D'_x of G'_x of length at most $k + 1$ is obtained as in the two items above. And then, using Lemmas 5-9, the components of $G \setminus G'_x$ can be added to it as above to obtain the desired $\{x, y\}$ -path-decomposition of G with length at most $k + 1$. ■

5 Further work

The next step would be to design a polynomial time exact algorithm (if it exists) to compute the pathlength of Outerplanar graphs. Note that the increase of the length (+1) in our approximation algorithm comes from the contiguous property. The Example of Figure 2 shows that we cannot avoid this increase if we keep the contiguous property. Moreover, the *LtR* property has been prove from a contiguous path-decomposition. Therefore, this proof needs also to be adapted for the exact case. An other question would be to know whether our algorithm for trees can be adapted to chordal graphs. Moreover, the complexity of computing the pathlength (or 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.*, 31(2):1217–1243, 2017.
- [2] Achim Blumensath and Bruno Courcelle. Monadic second-order definable graph orderings. *Log. Methods Comput. Sci.*, 10(1), 2014.
- [3] H. L. Bodlaender and F. V. Fomin. Approximation of pathwidth of outerplanar graphs. *J. Alg.*, 43(2):190–200, 2002.
- [4] D. Coudert, F. Huc, and J.-S. Sereni. Pathwidth of outerplanar graphs. *J. Graph Theory*, 55(1):27–41, 2007.
- [5] Y. Dourisboure and C. Gavoille. Tree-decompositions with bags of small diameter. *Discret. Math.*, 307(16):2008–2029, 2007.
- [6] F. F. Dragan, E. Köhler, and A. Leitert. Line-distortion, bandwidth and path-length of a graph. *Algorithmica*, 77(3):686–713, 2017.
- [7] Feodor F. Dragan and Arne Leitert. Minimum eccentricity shortest paths in some structured graph classes. *J. Graph Algorithms Appl.*, 20(2):299–322, 2016.
- [8] G. Ducoffe, S. Legay, and N. Nisse. On the complexity of computing treebreadth. *Algorithmica*, 82(6):1574–1600, 2020.
- [9] Maurice Herlihy, Fabian Kuhn, Srikanta Tirthapura, and Roger Wattenhofer. Dynamic analysis of the arrow distributed protocol. *Theory Comput. Syst.*, 39(6):875–901, 2006.
- [10] John E. Hopcroft and Robert Endre Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.

- [11] Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *42nd Annual Symposium on Foundations of Computer Science, FOCS*, pages 10–33. IEEE, 2001.
- [12] Adrian Kosowski, Bi Li, Nicolas Nisse, and Karol Suchan. k-chordal graphs: From cops and robber to compact routing via treewidth. *Algorithmica*, 72(3):758–777, 2015.
- [13] R. Krauthgamer and J. R. Lee. Algorithms on negatively curved spaces. In *47th Annual IEEE Symp. on Foundations of Computer Science (FOCS 2006)*, pages 119–132, 2006.
- [14] Burkhard Monien and Ivan Hal Sudborough. Min cut is np-complete for edge weighted trees. *Theor. Comput. Sci.*, 58:209–229, 1988.
- [15] J. B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.