Approximation of pathwidth of outerplanar graphs

Hans L. Bodlaender[*] Fedor V. Fomin[†]

Abstract

There exists a polynomial time algorithm to compute the pathwidth of outerplanar graphs [3], but the large exponent makes this algorithm impractical. In this paper, we give an algorithm, that given a biconnected outerplanar graph G, finds a path decomposition of G of pathwidth at most at most twice the pathwidth of G plus one. To obtain the result, several relations between the pathwidth of a biconnected outerplanar graph and its dual are established.

1 Introduction

Much research has been done to compute the pathwidth of graphs. The notion of pathwidth first appeared in the theory on graph minors by Robertson and Seymour [8], and is equivalent to several other graph theoretic notions, e.g., vertex separation number, interval thickness, node search number. See [2, 1] for overviews.

In this paper, we consider the problem to approximate the pathwidth of outerplanar graphs. In [3], it was shown that the pathwidth of graphs with bounded treewidth can be computed in polynomial time. As outerplanar graphs have treewidth two, we know that pathwidth is polynomial time computable for outerplanar graphs. However, the exponent in the running time of this algorithm is rather large — already one step in the algorithm requires to work with sets of size $O(n^{11})$. In this paper, we give a linear algorithm, which approximates the pathwidth of a 2-connected outerplanar

[*] Department of Computer Science, Utrecht University, P. O. Box 80.089, 3508 TB Utrecht, the Netherlands. Email: hansb@cs.uu.nl

[†] Centro de Modelamiento Matemático, Universidad de Chile, Santiago, Chile and Faculty of Mathematics and Mechanics, St.Petersburg State University, Bibliotechnaya sq.2, St.Petersburg, 198904, Russia. Email: fomin@gamma.math.spbu.ru

graph with multiplicative factor two. Our algorithm is based on structural results on the relation between the pathwidth of a 2-connected outerplanar graph and its dual, which are interesting in their own right. This 'dual' relation combining with the results of Ellis et al. [4] that the pathwidth of trees can be computed in linear time are the main ingredients of our algorithm. Also, we show how to construct the corresponding path decomposition in $O(n \log n)$ time.

In [5], Govindran et al. give an $O(n \log n)$ time algorithm for approximating the pathwidth of an outerplanar graph with a multiplicative factor of three. We improve upon this paper for biconnected outerplanar graphs.

2 Definitions and notations

We use the following notations: $G = (V, E)$ is an undirected and finite graph with vertex set V and the edge set E, assumed to be without self-loops or parallel edges. A *plane graph* is a particular drawing of a planar graph in the plane without crossings. An *outerplane graph* is a planar embedding of an outerplanar graph with every vertex on the exterior face. Edges of an outerplane graph that are not on the boundary of the exterior face are called *internal.* If $G = (V, E)$ is a plane graph then $G^* = (V^*, E^*)$ denotes its geometric dual. The *weak dual* of a plane graph G is the graph obtained from the dual G^* by deleting the vertex corresponding to the exterior face of G.

Observation. The weak dual of an outerplane graph is a forest.

Observation. The weak dual of a 2-connected outerplane graph is a tree.

The notion of pathwidth was introduced by Robertson and Seymour [8]. (See [2] and [7] for surveys.) A *path decomposition* of a graph $G = (V, E)$ is a sequence $(X_1, X_2, \ldots, X_r)$ of subsets of V, such that

- $\bigcup_{1 \le i \le r} X_i = V$.
- For all $\{v, w\} \in E$, there is an i, $1 \le i \le r$, with $v, w \in X_i$.
- For all $1 \le i_0 < i_1 < i_2 \le r$, $X_{i_0} \cap X_{i_2} \subseteq X_{i_1}$.

The *width* of path decomposition $(X_1, X_2, \ldots, X_r)$ is $\max_{1 \le i \le r} |X_i| - 1$. The *pathwidth* of a graph is the minimum width over its path decompositions.

The notion of treewidth is strongly related to the notion of pathwidth. In this paper, we use a variant of this notion, which we call *semi treewidth.*

A *semi tree decomposition* of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$, $\{X_i \mid i \in I\}$ a collection of subsets of V and $T = (I, F)$ a tree, such that

- $\bigcup_{i \in I} X_i = V$.
- For all $\{v, w\} \in E$, there is an $i \in I$ with $v, w \in X_i$, or there are nodes $i_0, i_1 \in I$ with $v \in X_{i_0}$, $w \in X_{i_1}$, and i_0 and i_1 are adjacent in T.
- For all $i_0, i_1, i_2 \in I$, if i_1 is on the path in T from i_0 to i_2, then $X_{i_0} \cap X_{i_2} \subseteq X_{i_1}$.

The *width* of a semi tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$, and the *semi treewidth* of a graph is the minimum width over its semi tree decompositions.

The definition of tree decomposition is obtained by replacing the second condition in the definition above by

- For all $\{v, w\} \in E$, there is an $i \in I$ with $v, w \in X_i$.

The treewidth of a graph is the minimum width over its tree decompositions. Notice that *path decomposition* of a graph G can be defined as a tree decomposition with a tree T being path.

3 Pathwidth of outerplane graphs

The main purpose of this section is to prove the following theorem.

Theorem 1. *Let G^* be a 2-connected outerplane graph without loops and multiple edges, and let G be the dual of G^*. Then:*

$$\mathrm{pw}(G) \leq \mathrm{pw}(G^*) \leq 2\,\mathrm{pw}(G) + 2.$$

Proof. By Lemma 4, $\mathrm{pw}(G) \leq \mathrm{pw}(G^*)$. By Lemma 5, there is a planar inner triangulation H^* of G^* such that $\mathrm{pw}(H) \leq \mathrm{pw}(G) + 1$. Notice that $\mathrm{pw}(G^*) \leq \mathrm{pw}(H^*)$. Applying Lemma 9 for H we have that $\mathrm{pw}(H^*) \leq 2\,\mathrm{pw}(H)$. So we have:

$$\mathrm{pw}(G) \leq \mathrm{pw}(G^*) \leq \mathrm{pw}(H^*) \leq 2\,\mathrm{pw}(H) \leq 2\,\mathrm{pw}(G) + 2.$$

□

The lemmas needed for the proof above will follow in the remainder of this section. We need the following fact about pathwidth of trees.

Theorem 2 (Ellis et al. [4]). *1. Every tree T of pathwidth $k+1$, $k \geq 1$, has a vertex u such that the forest $T \setminus \{u\}$ has at least three connected components of pathwidth $\geq k$;*

2. *For any tree T, $\mathrm{pw}(T) \leq k+1$, $k \geq 1$, if and only if there is a path P such that every connected component of the forest $T \setminus V(P)$ has pathwidth $\leq k$.*

Lemma 3. *Let $G = (V, E)$ be the dual and let $T = (V_T, E_T)$ be the weak dual of a 2-connected outerplane graph $G^* = (V^*, E^*)$. Then $\mathrm{pw}(T) + 1 = \mathrm{pw}(G)$.*

Proof. Let $v \in V$ be the vertex corresponding to the exterior face of G^*, *i.e.*, $T = G \setminus \{v\}$.

Let us prove that $\mathrm{pw}(G) \geq \mathrm{pw}(T) + 1$. Suppose that $\mathrm{pw}(T) = k$. Then by Theorem 2 there is a vertex $u \in V_T$ such that at least three branches at u have pathwidth $\geq k-1$. Let T_1, T_2, T_3 be these branches.

Let $(X_1, X_2, \ldots, X_r)$ be a path decomposition of G. In this path decomposition there are sets X_{i_j} containing at least k vertices of T_j, $j \in \{1,2,3\}$, as the pathwidth of each T_j is at least $k-1$. Without loss of generality, we suppose that $i_1 < i_2 < i_3$. Choose $a \in X_{i_1} \setminus X_{i_2}$ and $b \in X_{i_3} \setminus X_{i_2}$. Then X_{i_2} is an (a,b)-separator. But there are at least two vertex disjoint (a,b)-paths in G avoiding $V(T_2)$ (one passing through u and one through v.) Hence $\mathrm{pw}(T) \geq (k+2) - 1 = k+1$.

The proof of $\mathrm{pw}(G) \leq \mathrm{pw}(T) + 1$ is obvious because $T = G - v$. □

Lemma 4. *Let G^* be a 2-connected outerplane graph without loops and multiple edges. Then $\mathrm{pw}(G) \leq \mathrm{pw}(G^*)$.*

Proof. Let $T = (V_T, E_T)$ be the weak dual of G^*. By Lemma 3, $\mathrm{pw}(T) = \mathrm{pw}(G) - 1$. We now will show that $\mathrm{pw}(T) \leq \mathrm{pw}(G^*) - 1$, using induction to the pathwidth of T. In the case $\mathrm{pw}(T) = 1$ the result clearly holds.

Suppose now that for every 2-connected outerplane graph having weak dual of pathwidth $\leq k$, the lemma is correct.

Let G^* be a 2-connected outerplane graph and T be its weak dual with $\mathrm{pw}(T) = k+1$. Then by Theorem 2 there is a vertex $u \in V_T$ such that the graph $T \setminus \{v\}$ has at least three components T_1, T_2, T_3 of pathwidth $\geq k$. Let H_1, H_2, H_3 be the subgraphs of G^* having T_1, T_2, T_3 as weak duals. These graphs are 2-connected and outerplane and by induction hypothesis $\mathrm{pw}(H_i) \geq k+1$, $i \in \{1,2,3\}$.

Because G^* is a 2-connected outerplane graph without loops and multiple edges, the face of G^* corresponding to vertex u is bounded by a cycle C of length ≥ 3. See Figure 1 for a schematic diagram. Notice that for every $i \in \{1,2,3\}$ the subgraph H_i has some vertices of C and there is a path p_i in C from H_j to H_k avoiding H_i, $i \neq j \neq k$. Let $(X_1, X_2, \ldots, X_k)$ be a path decomposition of G. In this path decomposition there are sets X_{i_j} containing at least $k+2$ vertices of H_j, $j \in \{1,2,3\}$. We may assume, without loss of

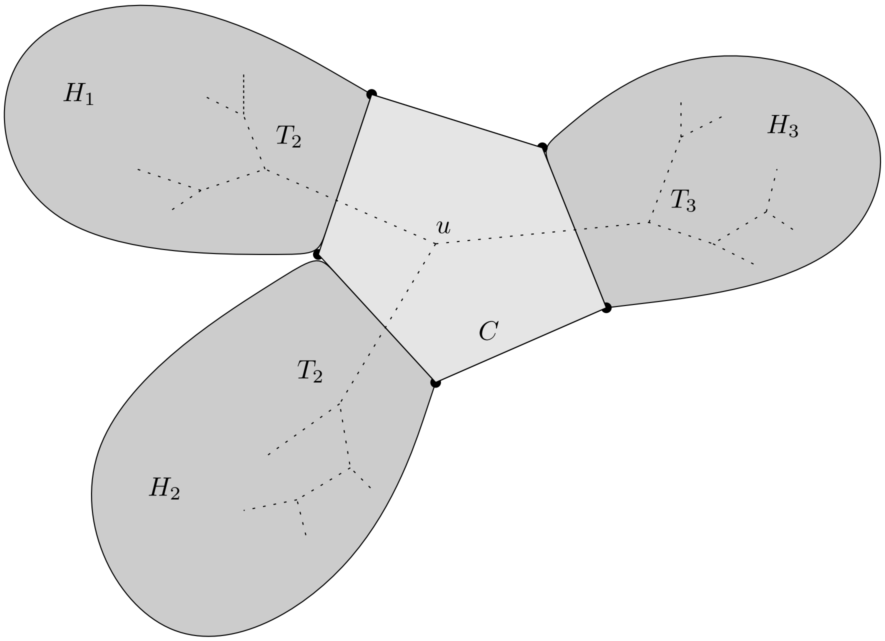

Figure 1: Illustration to the proof of Lemma 4

generality, that $i_1 < i_2 < i_3$. Then X_{i_2} separates X_{i_1} and X_{i_3}. Therefore X_{i_2} contains a vertex of p_2 and $|X_{i_2}| \geq k + 2 + 1 = k + 3$. □

A *triangulation* of a plane graph G is a maximal plane supergraph, *i.e.* a plane supergraph of G with every face (including the exterior face) a triangle.

An *inner triangulation* of a plane graph G is a plane supergraph of G, such that all interior faces are a triangle, i.e., we allow the exterior face not to be a triangle, and no edges are added to the outerface of G.

Lemma 5. *Let G^* be a 2-connected outerplane graph without loops and multiple edges. Then there exists an inner triangulation H^* of G^* such that* $\text{pw}(G) = \text{pw}(H) + 1$*, where $G = (G^*)^*$ and $H = (H^*)^*$.*

Proof. Consider the following *split* operation on plane graphs: take an interior face that is not a triangle, and add an edge between two non-adjacent vertices of the face. Clearly, when we repeat the split operation until it is no longer possible, we end up with an inner triangulation of the original graph. It remains to be shown that we can do this such that the pathwidth of the dual graphs do not change.

The split operation corresponds to the following operation on the dual G, or, similarly, to the weak dual T. Take a vertex v of degree at least 4 in G. Let $N[v]$ be the set of the vertices that are adjacent to v. Assume a clockwise ordering of the neighbors of v, $v_1, v_2, \ldots, v_s$. (The ordering is forced by the embedding, up to a cyclic shift.) Now, partition the set $N[v]$ into two sets M and N, where M consists of the vertices in some consecutive part of the ordering, i.e., $M = \{v_i, v_{i+1}, \ldots, v_j\}$, $1 \leq i < j \leq s$. M and N have size at least two. Now, transform G as follows: delete v with all its incident edges, add new vertices u and w with an edge $\{u, w\}$, and make u adjacent to all vertices of M and w adjacent to all vertices of N. Notice that the degree of u in new graph is $|N|+1$ and the degree of w is $|M|+1$. We say that the result of this transformation is obtained from G by a *vertex splitting* of v. A graph H is said to be a *split* of G if H is obtained from G by a sequence of vertex splittings. Notice that a split of an outerplane graph is an outerplane graph and if $F = (V_F, E_F)$ is a split of $G = (V, E)$ then $V_F^* = V^*$ and $E_F^* \supseteq E^*$.

Let T be a weak dual tree of G^*. We wish to prove that there is a split T_S of T such that $\mathrm{pw}(T) = \mathrm{pw}(T_S)$ and $\Delta(T_S) = 3$. By Lemma 3 this implies the existence of F^*, *i.e.* an outerplane graph having T_S as weak dual. By Theorem 2, every tree of pathwidth $\leq k+1$ has a path P such that every branch at this path (connected component of $T \setminus V(P)$) has pathwidth at most k. Using this fact one can find the split T_S easily. In fact, choose the path P as in Theorem 2 and split (if necessarily) the vertices of P. Since the new path has the same branches, such splittings do not increase the pathwidth. Then we can split the branches of P recursively, unless such a branch consists of a single vertex. Splitting a branch that is a single vertex means that the vertex is replaced by a path: this increases the pathwidth of the branch by one, and hence the pathwidth of G by one.

If we do the corresponding splits to G^*, then we obtain the desired inner triangulation of G^*. □

Note that a graph remains outerplanar when we apply an inner triangulation.

Lemma 6. *Let G_0 be a graph, and let G_1 be obtained from G_0 by removing all vertices of degree two whose neighbors are adjacent. Then* $\mathrm{pw}(G_1) \leq \mathrm{pw}(G_0) \leq \mathrm{pw}(G_1) + 1$.

Proof. As G_1 is a subgraph of G_0, clearly, $\mathrm{pw}(G_1) \leq \mathrm{pw}(G_0)$.

Suppose we have a path decomposition $(X_1, X_2, \ldots, X_r)$ of G_1. For each vertex v in G_0 with degree two whose neighbors w, x are adjacent, find a set X_i with $w, x \in X_i$ and add after X_i in sequence of the path decomposition a set $X_i \cup \{v\}$. This gives a path decomposition of G_0 whose width is at most one larger than the given path decomposition of G_1. □

Lemma 7. *Let H^* be a 2-connected inner triangulated outerplane graph. Let $T = (V_T, E_T)$ be the weak dual of H^*. Let H^- be the graph, obtained*

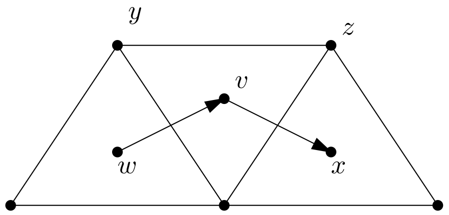

Figure 2: Illustration to the proof of Lemma 7

by removing all vertices of degree two from H^*. Then there is a semi tree decomposition $(\{X_i \mid i \in V_T\}, T)$ of H^- with width 1.

Proof. Choose an arbitrary leaf node v_0 from T, and view T as a rooted tree with root v_0. We take $X_{v_0} = \emptyset$. For all nodes $w \neq v_0$, consider the edge from w to its parent in T. This edge is dual (crosses in the diagram) an edge, say $\{y, z\}$ from H^*. Then take $X_w = \{y, z\}$.

We claim that $(\{X_i \mid i \in V_T\}, T)$ defined in this way is a semi tree decomposition of H^-. We will verify the second and third condition of semi tree decomposition; the first then follows directly from the second.

Consider an edge $\{y, z\}$ from H^-. If $\{y, z\}$ is an internal edge, then it crosses an edge say $\{v, w\}$ from T. Suppose v is the parent of w in T. Then, by construction, $\{y, z\} = X_w$. Now, suppose $\{y, z\}$ is not an internal edge. Suppose $\{y, z\}$ is adjacent to an internal face, represented by a vertex v from T. As the graph is triangulated, this face must also be a triangle. The face must be adjacent to two other internal faces in H^* otherwise y or z would have degree two in H^*, and hence $\{y, z\}$ would not be an edge in H^-. Say these faces are represented by nodes w and x. As the root of T is a node of degree one in T, either w or x is a child of v in T. Suppose w is a child of v in T. Then $y \in X_v$ and $z \in X_w$ or $z \in X_v$ and $y \in X_w$. (See diagram 2.)

Finally, for a node z in H^-, note that the edges from T that cross an edge with z as endpoint form a path in T. By the choice of a leaf node as root of T, this is a directed path and hence the collection of sets X_v that z induces a path in T. □

Lemma 8. *Let (X, T) be a semi tree decomposition of a graph $G = (V, E)$ with width k. Then,*

$$\mathrm{pw}(G) \leq (k+1)(\mathrm{pw}(T)+1) - 1.$$

Proof. Let $(Y_1, Y_2, \ldots, Y_r)$ be a path decomposition of T of pathwidth $\mathrm{pw}(T)$. Consider the sequence $(Z_1, Z_2, \ldots, Z_r)$ with $Z_i = \bigcup_{j \in Y_i} X_i$.

We now will verify that this sequence is a path decomposition of G. First, suppose $i_1 < i_2 < i_3$, and $v \in Z_{i_1} \cap Z_{i_3}$. Hence, $v \in X_{j_1}$ for some $j_1 \in Y_{i_1}$, and $v \in X_{j_3}$ for some $j_3 \in Y_{i_3}$. By the properties of path decomposition, we have that Y_{i_2} must contain a node from the path from j_1 to j_3 in T; call this node j_2. By the definition of semi tree decomposition, $v \in X_{j_2}$, and hence $v \in Z_{i_2}$.

Next, consider an edge $\{v, w\} \in E$. If there is an edge $\{i_1, i_2\}$ in T with $v \in X_{i_1}$ and $w \in X_{i_2}$, then there must be an j, $1 \leq j \leq r$ with $i_1, i_2 \in Y_j$. Hence, $v, w \in Z_j$. The case that there is an $i \in I$ with $v, w \in X_i$ is easy.

One directly sees that the width of this path decomposition is at most $(k+1)(\text{pw}(T)+1)-1$. □

Lemma 9. *Let G^* be a 2-connected inner triangulated outerplane graph with dual G. Then* $\text{pw}(G^*) \leq 2 \cdot \text{pw}(G)$.

Proof. Let H^- be the graph obtained by removing the nodes of degree two from H^*. Note that, as H^* is triangulated, the neighbors of every vertex of degree two in H^* are adjacent. By Lemma 6, $\text{pw}(H^*) \leq \text{pw}(H^-) + 1$. Let T be the weak dual of H^*. By Lemma 7, there is a semi tree decomposition (X, T) of H^- of width 1, hence by Lemma 8, $\text{pw}(H^-) \leq 2 \cdot \text{pw}(T) + 1$. Now

$$pw(H^*) \leq \text{pw}(H^-) + 1 \leq 2 \cdot \text{pw}(T) + 2 = 2 \cdot \text{pw}(H)$$

□

4 An approximation algorithm for biconnected outerplanar graphs

In this section, we give an algorithm, that given a 2-connected outerplanar graph $G^* = (V^*, E^*)$with n vertices finds a path decomposition of G^* that has a width that is at most two times the pathwidth of G^* plus 2. The algorithm uses $O(n \log n)$ time, and follows the structure of the proof, given in the previous section. Note that if we just want to have a bound on the pathwidth of G^*, then the algorithm can run in linear time.

Step 1: Remove loops and parallel edges. If we allow our input graph to have self-loops (edges of the form $\{v, v\}$, or parallel edges, then we just remove such edges: this does not change the pathwidth of the graph.

Step 2: Compute an outerplane embedding of G^*. It is well known that given a graph G, one can test in linear time if G is outerplanar, and if so, find an embedding of G with all vertices on the exterior face. See e.g. [6, 10].

Step 3: Compute the dual graph G of G^*. It is well known, that, given a plane embedding of a planar graph, one can find its dual in linear time.

Step 4: Compute the pathwidth of the dual graph G. The pathwidth of the dual graph G can be computed in linear time, in the following way. First, we take the tree T that is obtained from G by removing the vertex v representing the exterior face, i.e., T is the weak dual of G^*. Using the algorithm of Ellis et al. [4], we can compute in linear time the pathwidth of T, and compute a path decomposition $(X_1, \ldots, X_r)$ of T of optimal width in $O(n \log n)$ time. The pathwidth of G is one larger, by Lemma 3, and the path decomposition $(X_1 \cup \{v\}, X_2 \cup \{v\}, \ldots, X_r \cup \{v\})$ is a path decomposition of G of optimal width. By Euler formula the number of vertices in G is $O(n)$ and the pathwidth of G can be computed in $O(n)$ time.

If we are satisfied with a *bound* on the pathwidth of G^* that is at most two times plus 2 larger than the exact bound, then, by Theorem 1 we are done, as $2\,\mathrm{pw}(G) + 2$ is such a bound. However, if we want a *path decomposition* of G^* whose width at most two times plus 2 than the pathwidth of G^*, then more work has to be done.

Step 5: Compute an inner triangulation H^* of G^*, such that $\mathrm{pw}(H) = \mathrm{pw}(G) + 1$. In this step, we make the proof of Lemma 5 constructive, i.e., we compute an inner triangulation H^* of G^*, such that the pathwidth of G equals the pathwidth of H, where H is the dual of H^*.

Let T be again the weak dual of G^*. Suppose the pathwidth of T is k.

Now, first we find a path P in T, such that all branches at this path have pathwidth at most k. This can be done in linear time, using a minor modification of the algorithm from Ellis et al. [4]. When we have the path P, the split can easily be applied. After this, we continue recursively on the branches. As the pathwidth of a tree with t vertices is $O(\log t)$, every edge of T is involved in $O(\log t)$ recursive steps, and the total time of this step becomes $O(t \log t) = O(n \log n)$.

Step 6: Compute a path decomposition for the weak dual of H^* Compute the weak dual T of the inner triangulation H^*, computed in the previous step, and use the linear time algorithm of Ellis et. al. [4] to compute an optimal path decomposition of T.

Step 7: Compute the semi tree decomposition of H^- Let H^- be the graph obtained by removing the vertices of degree two from H^*. Following construction described in the proof of Lemma 7, we can make a semi tree decomposition (X, T) of width 1, with T the weak dual of H^*. It is not hard to see that this step can be done in linear time.

Step 8: Make a path decomposition of H^- The proof of Lemma 8 can be made constructive in a straightforward way; we thus obtain a path decomposition of H^- of width $2\,\mathrm{pw}(T) + 3 = 2\,\mathrm{pw}(H) + 1$ of H^-.

Step 9: Make a path decomposition of H^* Finally, we have to add back the vertices with degree two, while increasing the pathwidth with at most one, as in the proof of Lemma 6. This can be done in linear time by using the following method:

Suppose that the vertices in H^* are numbered $v_1, \ldots, v_n$. We use also a boolean array Z with indexes $\{1, \ldots, n\}$. Initially, all $Z[i]$ are false.

1. Suppose that S is the set of vertices of degree two. Make a set P of (ordered) pairs of vertices, with for every $v \in S$, its two neighbors as two ordered pairs in P, *i.e.*, if v's neighbors are x and y, then both (x, y) and (y, x) belong to P. Maintain pointers from and to a vertex $v \in S$ and its corresponding pair in P.

2. Radix sort P, *i.e.*, P is sorted with respect to the lexicographic ordering. This can be done in linear time, using the standard radix sort algorithm.

3. Now, we can build an array, with the ith entry pointing to a list of vertices v for which (v_i, v) belongs to P. This can be done in linear time, using the sorted list of the previous step.

4. Next, visit the bags in the path decomposition of H^- one by one. For each bag X_i, we do the following:

   (a) for every $v_j \in X_i$, make $Z[j]$ true.

   (b) for every $v_j \in X_i \backslash X_{i-1}$ and for every $v_{j'}$ with $(v_j, v_{j'}) \in P$, check if $Z[j']$ is true. If it is true, then we found the bag for the neighbors of a vertex v, namely the one corresponding to the pair $(v_j, v_{j'})$.

We remove this pair and its reversed copy $(v_{j'}, v_j)$ from the array of step 3, and add a pointer from v to bag X_i.

(c) for every $v_j \in X_i$, make $Z[j]$ false, i.e., now all values in Z are again false.

5. Now, for every vertex v of degree two in G^*, we can follow the pointer from v to a bag X_i. Add after X_i a bag which contains $X_i \cup \{v\}$. Note that in this way, we never create bags that are more than one larger than a bag in G^-.

When (v, w) and (w, v) belong to P, then consider the first bag that contains both v and w. In this bag, we either will consider v or w in step 4b, and hence the vertex of degree 2 associated with this pair will be pointing towards this bag.

The time is linear in the size of the path decomposition plus the size of G^*. The pathwidth of a tree with k vertices is $O(\log k)$ and by construction the number of vertices in a path decomposition is $O(n)$. Therefore this step can be performed in $O(n \log n)$ time.

Step 10: Obtain a path decomposition of G^* The path decomposition of H^* obtained in Step 9 is also the path decomposition of G^* and the width of this decomposition is at most $2\,\mathrm{pw}(G^*) + 2$.

5 Concluding remarks

One of the most interesting question about pathwidth of outerplanar graphs we left open is the existence of fast practical exact algorithms or algorithms approximating pathwidth of outerplanar graphs with additive factor. One of the possible ways in obtaining such algorithms is the proof of the following conjecture.

Conjecture 10. There is a constant c such that for any 2-connected outerplanar graph G^* without loops and multiple edges $\mathrm{pw}(G) \leq \mathrm{pw}(G^*) \leq \mathrm{pw}(G) + c$.

Moreover, we suggest that

Conjecture 11. For any 2-connected planar graph G^* without loops and multiple edges $\mathrm{pw}(G) - 1 \leq \mathrm{pw}(G^*) \leq \mathrm{pw}(G) + 1$.

Path decompositions of trees and of outerplanar graphs with n vertices can have $\Theta(n)$ bags X_i of size $\Theta(\log n)$, thus, when using a straightforward representation one may need already $\Omega(n \log n)$ time just for writing the output. However, more compact representations of path decompositions exists, e.g., mark for each vertex the first and last bag it belongs to, or one can use the equivalent notion of vertex separations. These representations have size linear in the number of vertices. As Skodinis [9] has shown that (with such representations) one can find an optimal path decomposition of a given tree in linear time, we conjecture that the algorithm of Section 4 can be made to run in linear time, but there are several unresolved matters in this, and we leave this as an open problem. As a side remark, we note that the algorithm of Govindran et al. [5] can be made to run in linear time, using Skodinis' algorithm and a corresponding representation of the path decompositions.

References

[1] H. L. Bodlaender, *Treewidth: Algorithmic techniques and results*, in Proceedings 22nd International Symposium on Mathematical Foundations of Computer Science, MFCS'97, Lecture Notes in Computer Science, volume 1295, I. Privara and P. Ruzicka, eds., Berlin, 1997, Springer-Verlag, pp. 19–36.

[2] ——, *A partial k-arboretum of graphs with bounded treewidth*, Theor. Comp. Sc., 209 (1998), pp. 1–45.

[3] H. L. Bodlaender and T. Kloks, *Efficient and constructive algorithms for the pathwidth and treewidth of graphs*, J. Algorithms, 21 (1996), pp. 358–402.

[4] J. A. Ellis, I. H. Sudborough, and J. Turner, *The vertex separation and search number of a graph*, Information and Computation, 113 (1994), pp. 50–79.

[5] R. Govindran, M. A. Langston, and X. Yan, *Approximating the pathwidth of outerplanar graphs*, Inform. Proc. Letters, 68 (1998), pp. 17–23.

[6] S. L. Mitchell, *Linear algorithms to recognize outerplanar and maximal outerplanar graphs*, Inform. Proc. Letters, 9 (1979), pp. 229–232.

[7] R. H. Möhring, *Graph problems related to gate matrix layout and PLA folding*, in Computational Graph Theory, Comuting Suppl. 7, E. Mayr, H. Noltemeier, and M. Sysło, eds., Springer Verlag, 1990, pp. 17–51.

[8] N. Robertson and P. D. Seymour, *Graph minors. I. Excluding a forest*, J. Comb. Theory Series B, 35 (1983), pp. 39–61.

[9] K. Skodinis, *Computing optimal linear layouts of trees in linear time*, Technical Report MIP-9904, Dept. of Computer Science, University of Passau, Passau, Germany, 1999.

[10] M. M. Sysło, *Characterisations of outerplanar graphs*, Disc. Math., 26 (1979), pp. 47–53.