Theoretical Computer Science 554 (2014) 119–134

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

2-Connecting outerplanar graphs without blowing up the pathwidth

Jasine Babu [a,*], Manu Basavaraju [b], L. Sunil Chandran [a], Deepak Rajendraprasad [a]

[a] *Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India*
[b] *Department of Informatics, University of Bergen, Norway*

ARTICLE INFO

Article history:
Received 11 September 2013
Accepted 27 April 2014
Available online 6 May 2014

Keywords:
Pathwidth
Outerplanar graph
2-Vertex-connected

ABSTRACT

Given a connected outerplanar graph G of pathwidth p, we give an algorithm to add edges to G to get a supergraph of G, which is 2-vertex-connected, outerplanar and of pathwidth $O(p)$. This settles an open problem raised by Biedl [1], in the context of computing minimum height planar straight line drawings of outerplanar graphs, with their vertices placed on a two-dimensional grid. In conjunction with the result of this paper, the constant factor approximation algorithm for this problem obtained by Biedl [1] for 2-vertex-connected outerplanar graphs will work for all outer planar graphs.

1. Introduction

A graph $G(V, E)$ is outerplanar, if it has a planar embedding with all its vertices lying on the outer face. Computing planar straight line drawings of planar graphs, with their vertices placed on a two-dimensional grid, is a well known problem in graph drawing. The height of a grid is defined as the smaller of the two dimensions of the grid. If G has a planar straight line drawing, with its vertices placed on a two-dimensional grid of height h, then we call it a planar drawing of G of height h. It is known that any planar graph on n vertices can be drawn on an $(n-1) \times (n-1)$ sized grid [2]. A well studied optimization problem in this context is to minimize the height of the planar drawing.

Pathwidth is a structural parameter of graphs, which is widely used in graph drawing and layout problems [1,3,4]. We use $\mathrm{pw}(G)$ to denote the pathwidth of a graph G. The study of pathwidth, in the context of graph drawings, was initiated by Dujmovic et al. [3]. It is known that any planar graph that has a planar drawing of height h has pathwidth at most h [4]. However, there exist planar graphs of constant pathwidth but requiring $\Omega(n)$ height in any planar drawing [5]. In the special case of trees, Suderman [4] showed that any tree T has a planar drawing of height at most $3\,\mathrm{pw}(T) - 1$. Biedl [1] considered the same problem for the bigger class of outerplanar graphs. For any 2-vertex-connected outerplanar graph G, Biedl [1] obtained an algorithm to compute a planar drawing of G of height at most $4\,\mathrm{pw}(G) - 3$. Since it is known that pathwidth is a lower bound for the height of the drawing [4], the algorithm given by Biedl [1] is a 4-factor approximation algorithm for the problem, for any 2-vertex-connected outerplanar graph. The method in Biedl [1] is to add edges to the 2-vertex-connected outerplanar graph G to make it a maximal outerplanar graph H and then draw H on a grid of height $4\,\mathrm{pw}(G) - 3$. The same method would give a constant factor approximation algorithm for arbitrary outerplanar graphs, if it were possible to add edges to an arbitrary connected outerplanar graph G to obtain a 2-vertex-connected outerplanar graph G' such that $\mathrm{pw}(G') = O(\mathrm{pw}(G))$. This was an open problem in Biedl [1].

* Corresponding author.
E-mail addresses: jasine@csa.iisc.ernet.in (J. Babu), iammanu@gmail.com (M. Basavaraju), sunil@csa.iisc.ernet.in (L.S. Chandran), deepakr@csa.iisc.ernet.in (D. Rajendraprasad).

http://dx.doi.org/10.1016/j.tcs.2014.04.032
0304-3975/

In this paper, we settle this problem by giving an algorithm to augment a connected outerplanar graph G of pathwidth p by adding edges so that the resultant graph is a 2-vertex-connected outerplanar graph of pathwidth $O(p)$. Notice that, the non-triviality lies in the fact that G' has to be maintained outerplanar. (If we relax this condition, the problem becomes very easy. It is easy to verify that the supergraph G' of G, obtained by making two arbitrarily chosen vertices of G adjacent to each other and to every other vertex in the graph, is 2-vertex-connected and has pathwidth at most $\mathrm{pw}(G)+2$.) Similar problems of augmenting outerplanar graphs to make them 2-vertex-connected, while maintaining the outerplanarity and optimizing some other properties, like number of edges added [6,7], have also been investigated previously.

2. Background

A *tree decomposition* of a graph $G(V,E)$ [8] is a pair $(T,\mathscr{X})$, where T is a tree and $\mathscr{X}=(X_t : t\in V(T))$ is a family of subsets of $V(G)$, such that:

1. $\bigcup(X_t : t\in V(T))=V(G)$.
2. For every edge e of G there exists $t\in V(T)$ such that e has both its end points in X_t.
3. For every vertex $v\in V$, the induced subgraph of T on the vertex set $\{t\in V(T) : v\in X_t\}$ is connected.

The width of the tree decomposition is $\max_{t\in V(T)}(|X_t|-1)$. Each $X_t\in\mathscr{X}$ is referred to as a bag in the tree decomposition. A graph G has *treewidth* w if w is the minimum integer such that G has a tree decomposition of width w.

A *path decomposition* $(P,\mathscr{X})$ of a graph G is a tree decomposition of G with the additional property that the tree P is a path. The width of the path decomposition is $\max_{t\in V(P)}(|X_t|-1)$. A graph G has *pathwidth* w if w is the minimum integer such that G has a path decomposition of width w.

Without loss of generality we can assume that, in any path decomposition $(P,\mathscr{X})$ of G, the vertices of the path P are labeled as $1,2,\ldots,$ in the order in which they appear in P. Accordingly, the bags in $\mathscr{X}$ also get indexed as $1,2,\ldots.$ For each vertex $v\in V(G)$, define $\mathit{FirstIndex}_{\mathscr{X}}(v)=\min\{i\mid X_i\in\mathscr{X}\text{ contains }v\}$, $\mathit{LastIndex}_{\mathscr{X}}(v)=\max\{i\mid X_i\in\mathscr{X}\text{ contains }v\}$ and $\mathit{Range}_{\mathscr{X}}(v)=[\mathit{FirstIndex}_{\mathscr{X}}(v),\mathit{LastIndex}_{\mathscr{X}}(v)]$. By the definition of a path decomposition, if $t\in\mathit{Range}_{\mathscr{X}}(v)$, then $v\in X_t$. If v_1 and v_2 are two distinct vertices, define $\mathit{Gap}_{\mathscr{X}}(v_1,v_2)$ as follows:

- If $\mathit{Range}_{\mathscr{X}}(v_1)\cap\mathit{Range}_{\mathscr{X}}(v_2)\neq\emptyset$, then $\mathit{Gap}_{\mathscr{X}}(v_1,v_2)=\emptyset$.
- If $\mathit{LastIndex}_{\mathscr{X}}(v_1)<\mathit{FirstIndex}_{\mathscr{X}}(v_2)$, then $\mathit{Gap}_{\mathscr{X}}(v_1,v_2)=[\mathit{LastIndex}_{\mathscr{X}}(v_1)+1,\mathit{FirstIndex}_{\mathscr{X}}(v_2)]$.
- If $\mathit{LastIndex}_{\mathscr{X}}(v_2)<\mathit{FirstIndex}_{\mathscr{X}}(v_1)$, then $\mathit{Gap}_{\mathscr{X}}(v_1,v_2)=[\mathit{LastIndex}_{\mathscr{X}}(v_2)+1,\mathit{FirstIndex}_{\mathscr{X}}(v_1)]$.

The motivation for this definition is the following. Suppose $(P,\mathscr{X})$ is a path decomposition of a graph G and v_1 and v_2 are two non-adjacent vertices of G. If we add a new edge between v_1 and v_2, a natural way to modify the path decomposition to reflect this edge addition is the following. If $\mathit{Gap}_{\mathscr{X}}(v_1,v_2)=\emptyset$, there is already an $X_t\in\mathscr{X}$, which contains v_1 and v_2 together and hence, we need not modify the path decomposition. If $\mathit{LastIndex}_{\mathscr{X}}(v_1)<\mathit{FirstIndex}_{\mathscr{X}}(v_2)$, we insert v_1 into all $X_t\in\mathscr{X}$, such that $t\in\mathit{Gap}_{\mathscr{X}}(v_1,v_2)$. On the other hand, if $\mathit{LastIndex}_{\mathscr{X}}(v_2)<\mathit{FirstIndex}_{\mathscr{X}}(v_1)$, we insert v_2 to all $X_t\in\mathscr{X}$, such that $t\in\mathit{Gap}_{\mathscr{X}}(v_1,v_2)$. It is clear from the definition of $\mathit{Gap}_{\mathscr{X}}(v_1,v_2)$ that this procedure gives a path decomposition of the new graph. Whenever we add an edge (v_1,v_2), we stick to this procedure to update the path decomposition.

A *block* of a connected graph G is a maximal connected subgraph of G without a cut vertex. Every block of a connected graph G is thus either a single edge which is a bridge in G, or a maximal 2-vertex-connected subgraph of G. If a block of G is not a single edge, we call it a non-trivial block of G. Given a connected outerplanar graph G, we define a rooted tree T (hereafter referred to as the *rooted block tree* of G) as follows. The vertices of T are the blocks of G and the root of T is an arbitrary block of G which contains at least one non-cut vertex (it is easy to see that such a block always exists). Two vertices B_i and B_j of T are adjacent if the blocks B_i and B_j share a cut vertex in G. It is easy to see that T, as defined above, is a tree. In our discussions, we restrict ourselves to a fixed rooted block tree of G and all the definitions hereafter will be with respect to this chosen tree. If block B_i is a child block of block B_j in the rooted block tree of G, and they share a cut vertex x, we say that B_i is a child block of B_j at x.

It is known that every 2-vertex-connected outerplanar graph has a unique Hamiltonian cycle [9]. Though the Hamiltonian cycle of a 2-vertex-connected block of G can be traversed either clockwise or anticlockwise, let us fix one of these orderings, so that the **successor** and **predecessor** of each vertex in the Hamiltonian cycle in a block is fixed. We call this order the clockwise order. Consider a non-root block B_i of G such that B_i is a child block of its parent, at the cut vertex x. If B_i is a non-trivial block and y_i and y_i' respectively are the predecessor and successor of x in the Hamiltonian cycle of B_i, then we call y_i the last vertex of B_i and y_i' the first vertex of B_i. If B_i is a trivial block, the sole neighbor of x in B_i is regarded as both the first vertex and the last vertex of B_i. By the term **path**, we always mean a simple path, i.e., a path in which no vertex repeats.