Computing β-Drawings of 2-Outerplane Graphs in Linear Time
(Extended Abstract)

Md. Abul Hassan Samee, Mohammad Tanvir Irfan, and Md. Saidur Rahman

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
{samee,mtirfan,saidurrahman}@cse.buet.ac.bd

Abstract. A straight-line drawing of a plane graph G is a drawing of G where each vertex is drawn as a point and each edge is drawn as a straight-line segment without edge crossings. A proximity drawing Γ of a plane graph G is a straight-line drawing of G with the additional geometric constraint that two vertices of G are adjacent if and only if no other vertex of G is drawn in Γ within a "proximity region" of these two vertices in Γ. Depending upon how the proximity region is defined, a given plane graph G may or may not admit a proximity drawing. In one class of proximity drawings, known as β-drawings, the proximity region is defined in terms of a parameter β, where $\beta \in [0, \infty)$. A plane graph G is β-drawable if G admits a β-drawing. A sufficient condition for a biconnected 2-outerplane graph G to have a β-drawing is known. However, the known algorithm for testing the sufficient condition takes time $O(n^2)$. In this paper, we give a linear-time algorithm to test whether a biconnected 2-outerplane graph G satisfies the known sufficient condition or not. This consequently leads to a linear algorithm for β-drawing of a wide subclass of biconnected 2-outerplane graphs.

Keywords: Graph Drawing, Proximity Drawing, β-Drawing, Proximity Graph, 2-Outerplane graph, Slicing Path, Good Slicing Path.

1 Introduction

Let Γ be a straight-line drawing of a plane graph G. Let $\Gamma(u)$ be the point on the plane to which the vertex u of G is mapped in Γ. Then Γ is a *proximity drawing* of the plane graph G if Γ satisfies the following proximity constraint: two vertices u and v of G are adjacent if and only if a well-defined "proximity region" corresponding to the points $\Gamma(u)$ and $\Gamma(v)$ is empty, i.e. the region does not contain $\Gamma(w)$ for any other vertex w of G. The exact definition of proximity region is problem-specific. As a matter of fact, there is an infinite number of different types of proximity regions. For example, an infinite family of parameterized proximity regions has been introduced in [5]. This family of parameterized proximity regions gives rise to an important class of proximity drawings, known as *β-drawings*, where β stands for a parameter that can take any real number value in $[0, \infty)$.

A plane graph G is β-drawable if G admits a β-drawing. Not all graphs are β-drawable for all values of β. For example, the graph G_1 illustrated in Fig. 1(a) is not

S.-i. Nakano and Md.S. Rahman (Eds.): WALCOM 2008, LNCS 4921, pp. 81–87, 2008.

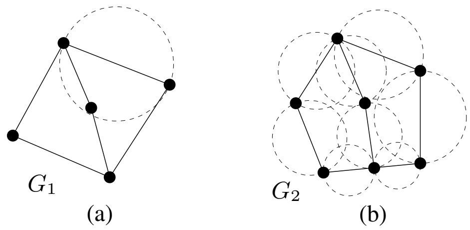

Fig. 1. (a) A graph G_1 which is not β-drawable for $\beta \in (1, 2)$, and (b) a graph G_2 which is β-drawable under the same constraints

β-drawable for $\beta = 1$. This fact can be explained as follows. Suppose we want to achieve a β-drawing of G_1 with $\beta = 1$. The β-region of two vertices u and v for $\beta = 1$ is a circle with $\Gamma(u)$ and $\Gamma(v)$ as its two antipodal points. For the graph G_1, wherever we place the four external vertices, the internal vertex will be inside the β-region of at least one of the four pairs of neighboring external vertices (as shown with the dotted circle in Fig. 1(a)). Therefore the graph G_1 is not β-drawable for $\beta = 1$. Following this same line of reasoning, one can easily work out that the graph G_2 shown in Fig. 1(b) is β-drawable for $\beta = 1$.

The *proximity drawability problem*, i.e. the problem whether a given graph admits a particular proximity drawing or not, has originated from the well-known "proximity graphs." Proximity graphs have wide applications in computer graphics, computational geometry, pattern recognition, computational morphology, numerical analysis, computational biology, GIS, instance-based learning and data-mining [3].

Several research outcomes regarding the proximity drawability of trees and outerplanar graphs are known [1, 2, 6]. One of the problems left open in [6] is to extend the problem of β-drawability of graphs to other nontrivial classes of graphs apart from trees and outerplanar graphs. In [4], the authors gave a sufficient condition for β-drawability of biconnected 2-outerplane graphs, where $\beta \in (1, 2)$. Although their sufficient condition induces a large and non-trivial class of biconnected 2-outerplane graphs, their algorithm for testing whether a given biconnected 2-outerplane graph satisfies those conditions or not, takes time $O(n^2)$.

In this paper, we give a linear-time algorithm for testing whether a biconnected 2-outerplane graph G satisfies the sufficient condition presented in [4]. Our algorithm essentially relies on the sufficient condition presented in [4], but works on a new set of conditions devised by us on "slicing paths" of G.

The rest of this paper is organized as follows. In Section 2, we present some definitions and preliminary results. In Section 3, we give a linear-time algorithm to test whether G satisfies the sufficient condition presented in [4] or not, and in the positive case, to compute a β-drawing of G. Finally, Section 4 is a conclusion.

2 Preliminaries

In this section we give some definitions and present our preliminary results.

A graph G is *connected* if there is a path between every pair of vertices of G, otherwise G is *disconnected*. A connected graph G is *biconnected* if at least two vertices of G are required to be removed to make the resulting graph disconnected. A *component* of G is a maximal connected subgraph of G. A graph G is *planar* if G has an embedding in the plane where no two edges cross each other, except at vertices on which two or more edges are incident. A *plane graph* is a planar graph with a fixed planar embedding. A plane graph partitions the plane into topologically connected regions called *faces*. The unbounded face is called the *external face*, while the remaining faces are called *internal faces*.

An *outerplanar graph* is a graph that has a planar embedding in which all the vertices lie on the external face. An outerplanar graph is also known as a 1-*outerplanar graph*. If all the vertices of a 1-outerplanar graph G appear on the external face of a given embedding of G, then we say that the embedded graph is a 1-*outerplane graph*, otherwise the embedded graph is not 1-outerplane, even though G is 1-outerplanar. These definitions can be generalized as follows. For an integer $k > 1$, an embedded graph is k-*outerplane* if the embedded graph obtained by removing all the vertices of the external face is a $(k-1)$-outerplane graph. On the other hand, we call a graph k-*outerplanar* if it has an embedding that is k-outerplane.

Let G be a biconnected 2-outerplane graph. The vertices on the external face of G are called the *external vertices* of G. The remaining vertices of G are called the *internal vertices* of G. For any external vertex u of G, the *fan of* u, denoted by F_u, is the subgraph of G induced by the vertices of G that share an internal face with u. The vertex u is called the *apex of* F_u.

For any two distinct points in the plane there is an associated region parameterized by β, where $\beta \in [0, \infty)$, which is called the β-region of the two points. There exist two variants of β-regions, namely lune-based and circle-based β-regions [5]. These regions can be further subdivided into two types: *open* β-*regions* and *closed* β-*regions*. In an open β-region, the boundary of the region is excluded from the region. However, in a closed β-region, the boundary of the region is included in the region. In [4], the authors studied the lune-based closed β-regions and gave the following sufficient condition for a biconnected 2-outerplane graph G to admit a β-drawing where $\beta \in (1, 2)$.

Theorem 1. *A biconnected* 2*-outerplane graph G is β-drawable for $\beta \in (1, 2)$ if G satisfies the following conditions 1 and 2.*

1. *There are at least five external vertices; and*
2. *There is an external vertex u such that the fan F_u has all of the following properties: (a) F_u is biconnected* 1*-outerplane; (b) F_u contains all the internal vertices of G and the internal vertices of G induces a single connected component of G; and (c) every vertex in F_u has at most one neighbor outside F_u and every vertex outside F_u has at most one neighbor in F_u.*

□

However, the sufficient conditions mentioned in Theorem 1 apply for lune-based open β-regions as well [4].

A constructive proof of Theorem 1 has been provided in [4]. In that proof, the authors have first tested whether a biconnected 2-outerplane graph G satisfies the conditions stated in Theorem 1 or not and in the positive case, they have computed a set of external vertices of G such that for each vertex u in that set, the corresponding fan F_u satisfies Conditions $2(a)$–$2(c)$ of Theorem 1. Such a set of external vertices is called the *set of candidate apices*. For the purpose of computing this set, the authors in [4] have adopted the following approach. For each external vertex u of G they have tested whether the fan F_u of u satisfies the Conditions $2(a)$–$2(c)$ of Theorem 1. For a specific fan F_u, this checking will take $O(n)$ time. Since there are $O(n)$ number of external vertices of G, this would require $O(n^2)$ time to compute the set of candidate apices. After computing the set of candidate apices, the authors in [4] have computed a β-drawing of G in time $O(n)$ as follows. They have first drawn the fan F_u of vertex u where u is a candidate apex. They have next drawn the remaining graph $G - (V(F_u))$ and have added edges between the vertices of F_u and the vertices of $G-(V(F_u))$. Thus the constructive proof of Theorem 1 presented in [4] yields an $O(n^2)$ time algorithm which finds candidate apices in $O(n^2)$ time and computes a β-drawing in $O(n)$ time.

In the remainder of this section, we present some fundamental observations which we use to compute the set of candidate apices in linear-time. We first have the following lemma whose proof is omitted in this extended abstract.

Lemma 1. *Let G be a biconnected 2-outerplane graph. Let u be an external vertex of G and F_u be the fan of vertex u. If F_u is 1-outerplane and F_u contains all the internal vertices of G then F_u is biconnected.* □

It is important here for us to mention the significance of Lemma 1 in our work. Throughout the remainder of this paper, given a biconnected 2-outerplane graph G, we will concentrate on an external vertex u of G and its fan F_u such that: *(a)* the condition given in Condition $2(a)$ of Theorem 1 stating that F_u should be 1-outerplane holds; and *(b)* Conditions $2(b)$ and $2(c)$ of Theorem 1 hold. Lemma 1 ensures that F_u will be biconnected in every such scenario, and hence, all the three conditions $2(a)$–$2(c)$ hold.

Let G be a biconnected 2-outerplane graph which satisfies the sufficient condition given in Theorem 1. We now have the following lemma regarding the subgraph G_{in} of G induced by the internal vertices of G. We have omitted the proof of Lemma 2 in this extended abstract.

Lemma 2. *Let $G = (V, E)$ be a biconnected 2-outerplane graph which satisfies the sufficient condition of Theorem 1. Let G_{in} be the subgraph of G induced by the internal vertices of G. Then G_{in} is a simple path.* □

3 Linear-Time Algorithm for β-Drawings of G

In this section, we first introduce the concept of a "slicing path" and a "good slicing path" of G. We next use the notion of good slicing paths of G to devise a linear-algorithm to test whether G satisfies the sufficient condition of Theorem 1 or not, and in the positive case, to compute a set of candidate apices of G. By using the linear-time algorithm presented in [4] for computing a β-drawing of such a graph G, we thus give a linear-time algorithm for computing a β-drawing of G.

Let G_{in} be the subgraph of G induced by the internal vertices of G. If G satisfies Theorem 1, then Lemma 2 implies that G_{in} is a simple path. Let $P_{st} = u_s, u_{s+1}, \ldots, u_t$, denote the path induced by the internal vertices of G. Clearly, every neighbor of u_s in G other than u_{s+1} is an external vertex of G. Similarly, every neighbor of u_t in G other than u_{t-1} is an external vertex of G. For an internal vertex v of G, let $N_{outer}(v)$ denote the number of those neighbors of v which are external vertices of G. We now have the following lemma.

Lemma 3. *Let G be a biconnected 2-outerplane graph which satisfies the sufficient condition of Theorem 1. Let $P_{st} = u_s, u_{s+1}, \ldots, u_t$, be the path induced by the internal vertices of G. Let $w \in \{u_s, u_t\}$. Then $N_{outer}(w) \leq 3$.* □

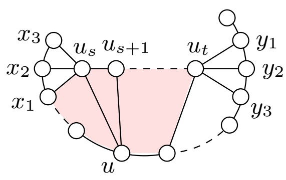

Fig. 2. Illustration of Lemma 3. The shaded region corresponds to the fan of vertex u

Proof. Let us assume that u is a candidate apex of G as illustrated in Fig. 2. Then F_u, the fan of u, will contain exactly one edge connecting u_s with an external vertex x of G where $x \neq u$. Similarly, F_u will contain exactly one edge connecting u_t with an external vertex y of G where $y \neq u$. Clearly, if $N_{outer}(w) > 3$ for either $w = u_s$ or $w = u_t$ as illustrated in Fig. 2, then w will have more than one neighbor outside F_u which violates Condition $2(c)$ of Theorem 1. Hence, $N_{outer}(w) \leq 3$, for each $w \in \{u_s, u_t\}$. □

Let G be a biconnected 2-outerplane graph which satisfies the sufficient condition of Theorem 1. Let the path $P_{st} = u_s, u_{s+1}, \ldots, u_t$, be the subgraph of G induced by the internal vertices of G. Then Lemma 3 holds for the end-vertices u_s and u_t of P_{st}. Let x and y be two of those external vertices of G which are neighbors of u_s and u_t respectively. We say that the path $P^*_{st} = x, u_s, u_{s+1}, \ldots, u_t, y$, is a *slicing path of G obtained from P_{st}*. Given a slicing path $P^*_{st} = x, u_s, u_{s+1}, \ldots, u_t, y$, if we traverse P^*_{st} from x to y, then some of the vertices of G will lie on our left hand side, and some will lie on our right hand side. Let G^L_{st} denote the subgraph of G induced by those vertices of G which lie on our left hand side while traversing P^*_{st} from x to y. Similarly, let G^R_{st} denote the subgraph of G induced by those vertices of G which lie on our right hand side while traversing P^*_{st} from x to y. We say that a slicing path $P^*_{st} = x, u_s, u_{s+1}, \ldots, u_t, y$, is a *good slicing path of G* if the following Conditions $(gs1)$ and $(gs2)$ hold.

$(gs1)$ There is no vertex v of P^*_{st} such that v has more than one neighbor in G^L_{st} and more than one neighbor in G^R_{st}; and

$(gs2)$ there are no two vertices u and v of P^*_{st} such that u and v have a common neighbor in G^L_{st} and a common neighbor in G^R_{st}.

The following lemma is immediate from the above definition of a good slicing path of G.

Lemma 4. *Let G be a biconnected 2-outerplane graph which satisfies the sufficient condition of Theorem 1. Let P_{st} be the path induced by the internal vertices of G. Then a good slicing path of G can be obtained from the path P_{st}.* □

We now have the following lemma.

Lemma 5. *Let G be a biconnected 2-outerplane graph which satisfies the sufficient condition of Theorem 1. Let $P_{st} = u_s, u_{s+1}, \ldots, u_t$, be the path induced by the internal vertices of G. In order to obtain a good slicing path of G from P_{st}, we need to check at most four slicing paths of G obtained from P_{st}.* □

Before presenting the proof of Lemma 5, we first give the following lemma whose proof is omitted in this extended abstract.

Lemma 6. *Let G be a biconnected 2-outerplane graph which satisfies the sufficient condition of Theorem 1. Let $P_{st} = u_s, u_{s+1}, \ldots, u_t$, be the path induced by the internal vertices of G. Let $N_{outer}(w) = 3$, for each $w \in \{u_s, u_t\}$. Let x_1, x_2, x_3 be the three neighbors of w which are external vertices of G appearing in counter-clockwise order on the external face of G. Then x_2 cannot be a candidate apex of G, but x_1 or x_3 can be a candidate apex of G.* □

Proof of Lemma 5. Let $w \in \{u_s, u_t\}$. We can obtain slicing paths P^*_{st} from P_{st} as follows. **(1)** If $N_{outer}(w) = 1$, then let x denote the neighbor of w which is an external vertex of G. In this case, we include x in P^*_{st}. **(2)** If $N_{outer}(w) = 3$, then let x_1, x_2, x_3 be the three neighbors of w appearing counter-clockwise on the external face of G. As we have shown in Lemma 6, x_2 cannot be a candidate apex of G, but x_1 or x_3 can be a candidate apex of G. In this case, we include x_2 in P^*_{st}. **(3)** If $N_{outer}(w) = 2$, then let x_1 and x_2 be the two neighbors of w on the external face of G. In this case, we construct two slicing paths from P_{st}, by including x_1 in one and including x_2 in another.

Hence, our claim holds from Lemma 6 and the above mentioned method to construct slicing paths P^*_{st} from P_{st}. □

We finally have the following lemma.

Lemma 7. *Let G be a biconnected 2-outerplane graph. Then one can check in linear time whether G satisfies the sufficient condition of Theorem 1 or not, and can compute the set of candidate apices of G in linear time if G satisfies the condition.*

Proof. Our proof is constructive. We first check whether G has at least five external vertices or not. We next check whether G_{in} is a simple path or not. If G_{in} is not a simple path, then Lemma 2 implies that G does not satisfy the sufficient condition of Theorem 1. In the positive case, let P_{st} denote the path induced by the internal vertices of G. Then we check whether Lemma 3 holds for P_{st} or not. If Lemma 3 does not hold for P_{st}, then Lemma 3 implies that G does not satisfy the sufficient condition of Theorem 1. Clearly, the operations in this first step can be performed in linear-time.

In our next step, we construct slicing paths P^*_{st} of G from P_{st} according to the method outlined in the proof of Lemma 5. It is also implied by Lemma 5 that we will have to construct at most four such slicing paths P^*_{st}. For each P^*_{st}, we check whether it is a good slicing path of G or not. If no good slicing path of G can be obtained from P_{st}, then we get from Lemma 4 that G does not satisfy the sufficient condition of Theorem 1. The checking of this step is independent of the previous step, and can be performed in linear-time. In this phase, we also remember which of the two subgraphs G^L_{st} and G^R_{st} can contain a candidate apex.

In our last step, for each good slicing path P^*_{st}, we traverse all the internal faces of G^L_{st} and G^R_{st} that can contain a candidate apex. Let u be an external vertex of G which appears on each of these faces. All such vertices u will constitute the set of candidate apices. This step can also be implemented in linear time, since we can gather the whole information by traversing each such internal face at most once. Hence the total algorithm has linear time complexity. □

We finally present the main result of this paper in the following theorem.

Theorem 2. *Let G be a biconnected 2-outerplane graph. Then one can check in linear time whether G satisfies the sufficient condition of Theorem 1 or not, and can find a β-drawing of G, where $\beta \in (1, 2)$, in linear time if G satisfies the condition.*

Proof. The claim holds directly from Lemma 7 and the linear-time drawing algorithm presented in [4]. □

4 Conclusion

In this paper, we have given a linear-time algorithm to test whether a biconnected 2-outerplane graph G satisfies the sufficient condition presented in [4], and thus, we have achieved a linear algorithm for computing β-drawings of biconnected 2-outerplane graphs where $\beta \in (1, 2)$. It remains as our future work to obtain efficient algorithms for computing β-drawings of larger classes of graphs.

References

[1] Bose, P., Di Battista, G., Lenhart, W., Liotta, G.: Proximity constraints and representable trees. In: Tamassia, R., Tollis, I(Y.) G. (eds.) GD 1994. LNCS, vol. 894, pp. 340–351. Springer, Heidelberg (1995)

[2] Bose, P., Lenhart, W., Liotta, G.: Characterizing proximity trees. Algorithmica 16, 83–110 (1996)

[3] Tamassia, R.(ed.): Handbook of graph drawing and visualization. CRC Press, Boca Raton, USA(to be published), `http://www.cs.brown.edu/~rt/gdhandbook/`

[4] Irfan, M.T., Rahman, M.S.: Computing β-drawings of 2-outerplane graphs. In: Kaykobad, M., Rahman, M. S. (eds.) WALCOM. Bangladesh Academy of Sciences (BAS), pp. 46–61 (2007)

[5] Kirkpatrick, D.G., Radke, J.D.: A framework for computational morphology. In: Toussaint, G.T. (ed.) Computational Geometry, pp. 217–248. North-Holland, Amsterdam (1985)

[6] Lenhart, W., Liotta, G.: Proximity drawings of outerplanar graphs. In: North, S.C. (ed.) GD 1996. LNCS, vol. 1190, pp. 286–302. Springer, Heidelberg (1997)