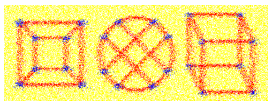

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 14, no. 1, pp. 19–51 (2010)

An Algorithm to Construct Greedy Drawings of Triangulations

Patrizio Angelini[1] *Fabrizio Frati*[1] *Luca Grilli*[2]

[1]Dipartimento di Informatica e Automazione,
Roma Tre University, Italy
[2]Dipartimento di Ingegneria Elettronica e dell'Informazione,
Perugia University, Italy

Abstract

We show an algorithm to construct a greedy drawing of every given triangulation. The algorithm relies on two main results. First, we show how to construct greedy drawings of a fairly simple class of graphs, called triangulated binary cactuses. Second, we show that every triangulation can be spanned by a triangulated binary cactus.

Further, we discuss how to extend our techniques in order to prove that every triconnected planar graph admits a greedy drawing. Such a result, which proves a conjecture by Papadimitriou and Ratajczak, was independently shown by Leighton and Moitra.

Submitted:	Reviewed:	Accepted:	Final:	Published:
November 2008	October 2009	October 2009	November 2009	January 2010
	Article type:		Communicated by:	
	Regular paper		I. G. Tollis and M. Patrignani	

Work partially supported by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB "Advanced tracking system in intermodal freight transportation".

E-mail addresses: angelini@dia.uniroma3.it (Patrizio Angelini) frati@dia.uniroma3.it (Fabrizio Frati) luca.grilli@diei.unipg.it (Luca Grilli)

1 Introduction

The standard Internet routing protocol is as follows: Each computer is univocally identified by an *IP-address*; IP-addresses are *aggregated*, i.e., computers that are topologically or geographically close in the network are assigned addresses with the same most significative bits; consequently, routers do not have to know the route to each address in the network, but they maintain in their *routing tables* only information on the route to take for reaching each set of aggregated addresses. Such an approach does not work in many wireless networks, such as *ad-hoc* and *sensor networks*, where the addresses that are assigned to nodes geographically or topologically close are not necessarily similar. Despite of their importance, no universally-accepted communication protocol exists for such wireless environments.

Geographic routing is a class of routing protocols in which nodes forward packets based on their geographic locations. Among such protocols, *geometric routing*, or *greedy routing*, has been well investigated, because it relies on a very simple strategy in which, in order to forward packets, each node has to know only local information and, obviously, the destination address. In fact, in the greedy routing a node forwards packets to a neighbor that is *closer than itself* to the destination's geographic location. Different distance metrics define different meanings for the word "closer" and consequently define different routing algorithms for the packet delivery. The most used and studied metric is of course the *Euclidean distance*.

The efficiency of the geographic routing algorithms strongly relies on the geographic coordinates of the nodes. This is indeed a drawback of such routing algorithms, for the following reasons: (i) Nodes of the network have to know their locations, hence they have to be equipped with GPS devices, which are expensive and increase the energy consumption of the nodes; (ii) geographic coordinates are independent of the network obstructions, i.e. obstacles making the communication between two close nodes impossible, and, more in general, they are independent of the network topology; this could lead to situations in which the communication fails because a *void* has been reached, i.e., the packet has reached a node whose neighbors are all farther from the destination than the node itself.

A brilliant solution to the geographic routing weakness has been proposed by Rao, Papadimitriou, Shenker, and Stoica, who in [16] proposed a scheme in which nodes are assigned *virtual coordinates* and then apply the standard geometric routing algorithm relying on such virtual locations rather than on the real geographic coordinates. Clearly, virtual coordinates do not need to reflect the nodes actual positions, hence they can be suitably chosen to guarantee that the geometric routing algorithm delivers packets with high probability. It has been experimentally shown that such an approach strongly improves the reliability of geometric routing [16, 15]. Further, it has been proved that virtual coordinates guarantee geometric routing to work for every connected topology when they can be chosen in the hyperbolic plane [11], even if only a logarithmic number of bits are available to store the coordinates of each node [7]. Moreover,

some easy modifications of the routing algorithm guarantee that Euclidean virtual coordinates can be chosen so that the packet delivery always succeeds [3], even if the coordinates need to be computed locally [4].

Subsequent to the publication of Rao *et al.* paper [16], an intense research effort has been devoted to determine on which network topologies the Euclidean geometric routing with virtual coordinates is guaranteed to work. From a graph-theoretic point of view, the problem can be restated as follows: Which are the graphs that admit a *greedy drawing*, i.e., a straight-line drawing Γ in the plane such that, for every pair of nodes u and v, there exists a *distance-decreasing path* in Γ? A path $(v_0, v_1, \ldots, v_m)$ is distance-decreasing if $d(v_i, v_m) < d(v_{i-1}, v_m)$, for $i = 1, \ldots, m$, where $d(p, q)$ denotes the Euclidean distance between two points p and q.

In [15] Papadimitriou and Ratajczak conjectured the following:

Conjecture 1 (Papadimitriou and Ratajczak [15]) *Every triconnected planar graph admits a greedy drawing.*

Papadimitriou and Ratajczak showed that $K_{k,5k+1}$ has no greedy drawing, for $k \geq 1$. As a consequence, both the triconnectivity and the planarity are necessary, because there exist planar non-triconnected graphs, such as $K_{2,11}$, and non-planar triconnected graphs, such as $K_{3,16}$, that do not admit any greedy drawing. Further, they observed that, if a graph G has a greedy drawing, then any graph containing G as a spanning subgraph has a greedy drawing, as well. It follows that Conjecture 1 extends to all graphs which are spanned by a triconnected planar graph. Related to such an observation, Papadimitriou and Ratajczak proved that every triconnected graph that does not have a $K_{3,3}$-minor has a triconnected planar spanning subgraph.

There are a few classes of triconnected planar graphs for which the conjecture is easily shown to be true, for example graphs with a *Hamiltonian path* and *Delaunay Triangulations*. At the *Symposium on Discrete Algorithms 2008* [6], Dhandapani proved the conjecture for the first non-trivial class of triconnected planar graphs, namely he showed that every *triangulation* admits a greedy drawing. Triangulations are clearly an important graph class to study, as also remarked by Papadimitriou and Ratajczak [15]. The proof of Dhandapani is probabilistic, namely the author proves that, for every given triangulation G, a *random Schnyder drawing* of G [17] is greedy with positive probability; hence, there exists a greedy drawing of every triangulation. Although such a proof is elegant, relying at the same time on an old Combinatorial Geometry theorem, known as the *Knaster-Kuratowski-Mazurkievicz Theorem* [12], and on standard Graph Drawing techniques, as the *Schnyder realizers* [17] and the *canonical orderings* of a triangulation [5], it does not lead to an drawing algorithm.

In this paper we show an algorithm for constructing greedy drawings of triangulations. The algorithm relies on a different and maybe easier approach with respect to the one used by Dhandapani. We define a simple class of graphs, called *triangulated binary cactuses*, and we provide an algorithm to construct a greedy drawing of any such a graph. Further, we show how to find, for every

triangulation, a binary cactus spanning it. It is clear that the previous statements imply an algorithm for constructing greedy drawings of triangulations. Namely, consider any triangulation G, apply the algorithm to find a binary cactus S spanning G, and then apply the algorithm to construct a greedy drawing of S. As already observed in [15], adding edges to a greedy drawing leaves the drawing greedy, hence S can be augmented to G, obtaining the desired greedy drawing of G.

Theorem 1 *Given a triangulation G, there exists an algorithm to compute a greedy drawing of G.*

Further, we provide an algorithm to construct greedy drawings of general triconnected planar graphs. The strategy of such an algorithm is the same as the one of the algorithm for constructing greedy drawings of triangulations. In fact, we define a simple class of graphs, called *non-triangulated binary cactuses*, and we provide an algorithm to construct a greedy drawing of any such a graph. Finally, we show how to find, for every triconnected planar graph, a non-triangulated binary cactus spanning it. Such a result proves Conjecture 1; however, the conjecture has been very recently (and independently) proved by Leighton and Moitra [13], by using techniques which are amazingly similar to ours. Hence, we will only sketch how to modify the algorithm we provide for triangulations in order to make it work for general triconnected planar graphs; however, we will extensively discuss differences and similarities of our algorithm with respect to Leighton and Moitra's one.

The rest of the paper is organized as follows. In Sect. 2 we establish some definitions and give some preliminaries. In Sect. 3 we show an algorithm to construct greedy drawings of triangulated binary cactuses. In Sect. 4 we show an algorithm to construct a triangulated binary cactus spanning a given triangulation. In Sect. 5 we show how to modify the algorithm described for triangulations in order to make it work for general triconnected planar graphs and we compare our techniques with Leighton and Moitra's ones; in the same section we conclude by providing some interesting open problems concerning greedy graph drawings.

A preliminary version of this paper appeared at *Graph Drawing 2008* [2].

2 Preliminaries

A graph $G'(V', E')$ is a *subgraph* of a graph $G(V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. A subgraph $G'(V', E')$ of a graph $G(V, E)$ is *induced* by V' if, for every edge $(u, v) \in E$ such that $u, v \in V'$, $(u, v) \in E'$. A graph $G'(V', E')$ is a *spanning subgraph* of $G(V, E)$ if it is a subgraph of G and $V' = V$.

A graph is *connected* if every pair of vertices of G is connected by a path. A *k-connected* graph G is such that removing any $k-1$ vertices leaves G connected; 3-connected, 2-connected, and 1-connected graphs are also called *triconnected*, *biconnected*, and *simply connected* graphs, respectively. A *separating k-set* is a set of k vertices whose removal disconnects the graph. Separating 1-sets and

separating 2-sets are also called *cutvertices* and *separation pairs*, respectively. Hence, a connected graph is biconnected if it has no cutvertices and it is triconnected if it has no separation pairs. The maximal biconnected subgraphs of a graph are its *blocks*. Each edge of G falls into a single block of G, while cutvertices are shared by different blocks. The *block-cutvertex tree*, or BC-tree, of a connected graph G is a tree with a B-node for each block of G and a C-node for each cutvertex of G. Edges in the BC-tree connect each B-node μ to the C-nodes associated with the cutvertices belonging to the block of μ.

The BC-tree of G may be thought as rooted at a specific block ν. When the BC-tree $\mathcal{T}$ of a graph G is rooted at a certain block ν, we denote by $G(\mu)$ the subgraph of G induced by all the vertices in the blocks contained in the subtree of $\mathcal{T}$ rooted at μ. In a rooted BC-tree $\mathcal{T}$ of a graph G, for each B-node μ we denote by $r(\mu)$ the cutvertex of G parent of μ in $\mathcal{T}$. If μ is the root of $\mathcal{T}$, i.e., $\mu = \nu$, then we let $r(\mu)$ denote any non-cutvertex node of the block associated with μ. In the following, unless otherwise specified, each considered BC-tree is meant to be rooted at a certain B-node ν such that the block associated with ν has at least one vertex $r(\nu)$ which is not a cutvertex. It is not difficult to see that such a block exists in every planar graph.

A *triangulated binary cactus* S, in the following simply called *binary cactus*, is a connected graph such that (see Fig 1):

- The block associated with each B-node of $\mathcal{T}$ is either an edge or a *triangulated cycle*, i.e., a cycle $(r(\mu), u_1, u_2, \ldots, u_h)$ triangulated by the edges from $r(\mu)$ to each of $u_2, u_3, \ldots, u_{h-1}$.
- Every cutvertex is shared by exactly two blocks of S.

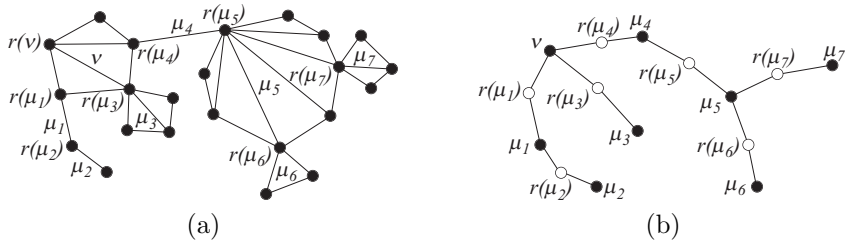


Figure 1: (a) A binary cactus S. (b) The block-cutvertex tree of S. White (resp. black) circles represent C-nodes (resp. B-nodes).

A *planar drawing* of a graph is a mapping of each vertex to a distinct point of the plane and of each edge to a Jordan curve between its endpoints such that no two edges intersect except, possibly, at common endpoints. A *straight-line drawing* is such that all the edges are straight-line segments. A planar drawing of a graph determines a circular ordering of the edges incident to each vertex. Two drawings of the same graph are *equivalent* if they determine the same circular ordering around each vertex. A *planar embedding* is an equivalence class

of planar drawings. A planar drawing partitions the plane into topologically connected regions, called *faces*. The unbounded face is the *outer face*. The outer face of a graph G is denoted by $f(G)$. A *chord* of a graph G is an edge connecting two non-consecutive vertices of $f(G)$. A graph together with a planar embedding and a choice for its outer face is called *plane graph*. A plane graph is a *triangulation* when all its faces are triangles. A plane graph is *internally-triangulated* when all its internal faces are triangles. An *outerplane* graph is a plane graph such that all its vertices are incident to the outer face. A *Hamiltonian cycle* of a graph G is a simple cycle passing through all vertices of G. Notice that a biconnected outerplane graph has only one Hamiltonian cycle, namely the one delimiting its outer face.

3 Greedy Drawing of a Binary Cactus

In this section, we give an algorithm to compute a greedy drawing of a binary cactus S. Such a drawing is constructed by performing a bottom-up traversal of the BC-tree $\mathcal{T}$ of S.

Consider the root μ of a subtree of $\mathcal{T}$ corresponding to a block of S, consider the k children of μ, which correspond to cutvertices of S, and consider the children of such cutvertices, say $\mu_1, \mu_2, \ldots, \mu_k$. Notice that each C-node child of μ is parent of exactly one B-node μ_i of $\mathcal{T}$, by the definition of binary cactus. For each $i = 1, \ldots, k$, inductively assume to have a drawing Γ_i of $S(\mu_i)$ satisfying the properties listed below.

Let C be a circle, let (a_i, b_i) be an arc of C, let p_i^* be a point of C such that the diameter through p_i^* cuts (a_i, b_i) in two arcs of the same length. Let α_i and β_i be any two angles such that $\alpha_i \leq \beta_i \leq \frac{\pi}{4}$. Consider the tangent $t(p_i^*)$ to C in p_i^*. Consider two half-lines l_1^* and l_2^* incident to p_i^*, lying on the opposite part of C with respect to $t(p_i^*)$, and forming angles equal to β_i with $t(p_i^*)$. Denote by $W(p_i^*)$ the wedge centered at p_i^*, delimited by l_1^* and l_2^*, and not containing C. Refer to Fig. 2.a.

- *Property 1.* Γ_i is a greedy drawing.
- *Property 2.* Γ_i is entirely contained inside a region $R(\Gamma_i)$ delimited by arc (a_i, b_i) and by segments $\overline{p_i^* a_i}$ and $\overline{p_i^* b_i}$. The angle $\widehat{a_i p_i^* b_i}$ is α_i.
- *Property 3.* For every vertex v in $S(\mu_i)$ and for every point p internal to $W(p_i^*)$, there exists in Γ_i a path $(v = v_0, v_1, \ldots, v_l = r(\mu_i))$ from v to $r(\mu_i)$ such that $d(v_j, p) < d(v_{j-1}, p)$, for $j = 1, \ldots, l$.
- *Property 4.* For every vertex v in $S(\mu_i)$ and for every point p internal to $W(p_i^*)$, $d(v, p_i^*) < d(v, p)$.

In the base case, block μ has no child. Denote by $(r(\mu) = u_0, u_1, \ldots, u_{h-1})$ the block of S corresponding to μ. Notice that $h \geq 2$. Consider any circle C with center c. Let p^* be the point of C with smallest y-coordinate. Consider

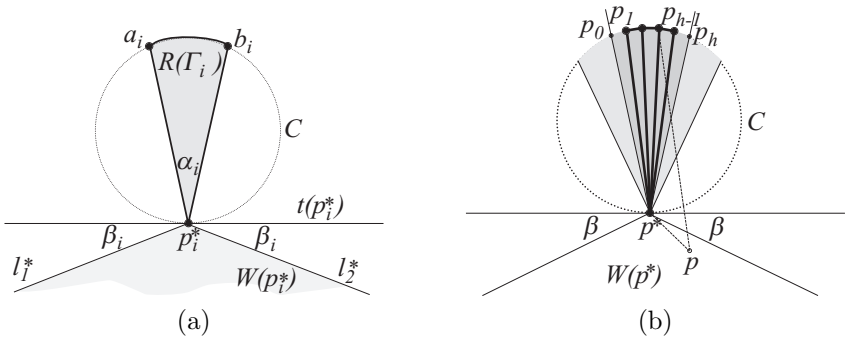


Figure 2: (a) Illustration for Properties 1–4 of Γ_i. (b) Base case of the algorithm. The light and dark shaded region represents $R(\Gamma)$ (the angle of $R(\Gamma)$ at p^* is α). The dark shaded region represents the intersection of $W(p^*, \frac{\alpha}{2})$ with the disk delimited by C.

the wedges $W(p^*, \alpha)$ and $W(p^*, \frac{\alpha}{2})$ with angles α and $\frac{\alpha}{2}$, respectively, incident to p^* and such that the diameter of C through p^* is their bisector (see Fig. 2.b). Place $r(\mu)$ at p^*. Denote by p'_a and p'_b the intersection points (different from p^*) of the half-lines delimiting $W(p^*, \frac{\alpha}{2})$ with C. Denote by A the arc of C between p'_a and p'_b and not containing p^*. Consider $h+1$ points $p_0, p_1, \ldots, p_h$ on A such that $p_0 = p'_a$, $p_h = p'_b$, and the distance between any two consecutive points p_i and p_{i+1} is the same. Place vertex u_i at point p_i, for $i = 1, 2, \ldots, h-1$. Notice that, if $h = 2$, μ corresponds to an edge of S that is drawn as a vertical segment, with u_1 above u_0.

In order to show that the constructed drawing Γ satisfies Property 1, consider any two vertices u_i and u_j, with $i < j$. If $i = 0$, then u_0 and u_j are joined by an edge, which provides a distance-decreasing path between them. Otherwise, we prove that $(u_i, u_{i+1}, \ldots, u_j)$ is a distance-decreasing path from u_i to u_j, the proof that $(u_j, u_{j-1}, \ldots, u_i)$ is a distance-decreasing path from u_j to u_i being analogous. For each $l = i, i+1, \ldots, j-2$, angle $\widehat{u_l u_{l+1} u_j}$ is greater than $\frac{\pi}{2}$, because triangle (u_l, u_{l+1}, u_j) is inscribed in less than half a circle with u_{l+1} as middle point (see Fig. 3.a). Hence, (u_l, u_j) is the longest side of triangle (u_l, u_{l+1}, u_j) and $d(u_{l+1}, u_j) < d(u_l, u_j)$ follows. Drawing Γ satisfies Property 2 by construction. In order to prove that Γ satisfies Property 3, we have to prove that, for every vertex u_i, with $i \geq 1$, and for every point p in $W(p^*)$, $d(u_0, p) < d(u_i, p)$. Angle $\widehat{pp^*p_i}$ is at least $\beta + (\frac{\pi}{2} - \frac{\alpha}{4})$, which is greater than $\frac{\pi}{2}$ (see Fig. 3.b). It follows that segment $\overline{pp_i}$ is the longest side of triangle (p, p^*, p_i), thus proving that $d(u_0, p) < d(u_i, p)$. For the same reason $d(u_0, u_i) < d(p, u_i)$, hence proving Property 4.

Now we discuss the inductive case. Suppose that μ is a node of $\mathcal{T}$ having k children. We show how to construct a drawing Γ of $S(\mu)$ satisfying Properties 1–4 with parameters α and β. Refer to Fig. 4. Denote by $(r(\mu) = u_0, u_1, \ldots, u_{h-1})$ the block of S corresponding to μ. Remember that $h \geq 2$ and that if $h = 2$, then the block is an edge, otherwise it is a triangulated cycle. Consider any circle C

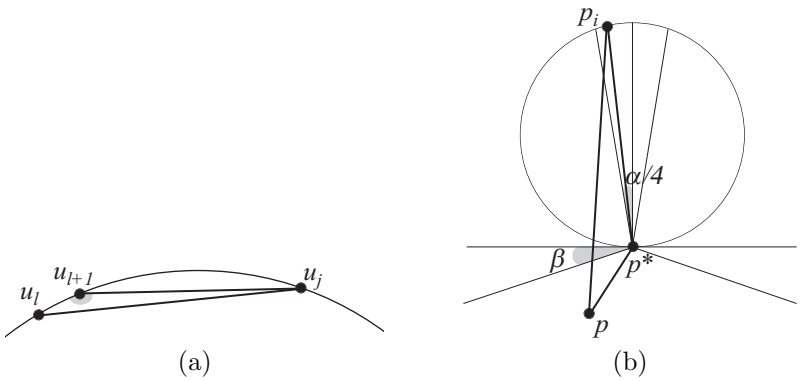


Figure 3: (a) Γ satisfies Property 1. (b) Γ satisfies Properties 3 and 4.

with center c. Let p^* be the point of C with smallest y-coordinate. Consider the wedges $W(p^*, \alpha)$ and $W(p^*, \frac{\alpha}{2})$ with angles α and $\frac{\alpha}{2}$, respectively, incident to p^* and such that the diameter of C through p^* is their bisector. Region $R(\Gamma)$ is the intersection region of $W(p^*, \alpha)$ with the closed disk delimited by C.

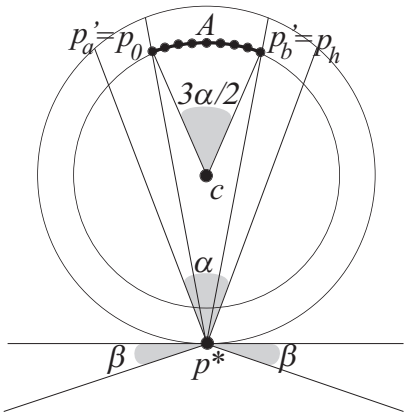


Figure 4: Construction of a drawing Γ in the inductive case of the algorithm.

Consider a second circle C' with center c intersecting the two lines delimiting $W(p^*, \frac{\alpha}{2})$ in two points p'_a and p'_b such that angle $\widehat{p'_a c p'_b} = \frac{3\alpha}{2}$. It is not difficult to see that such a circle always exists. Denote by A the arc of C' delimited by p'_a and p'_b and farther from p^*. Consider $h+1$ points $p_0, p_1, \ldots, p_h$ on A such that $p_0 = p'_a$ and $p_h = p'_b$, and the distance between any two consecutive points p_i and p_{i+1} is the same. Observe that, for each $i = 0, 1, \ldots, h-1$, angle $\widehat{p_i c p_{i+1}} = \frac{3\alpha}{2h}$.

First, we draw the block of S corresponding to μ. As in the base case, place vertex $u_0 = r(\mu)$ at p^* and, for $i = 1, 2, \ldots, h-1$, place u_i at point p_i. Recursively construct a drawing Γ_i of $S(\mu_i)$ satisfying Properties 1–4 with

$\alpha_i = \frac{3\alpha}{16h}$ and $\beta_i = \frac{3\alpha}{8h}$.

We are going to place each drawing Γ_i of $S(\mu_i)$ together with the constructed drawing of the block of S corresponding to μ, thus obtaining a drawing Γ of $S(\mu)$. Notice that not all the h nodes u_i are cutvertices of S. However, with a slight abuse of notation, we suppose that block $S(\mu_i)$ has to be placed at node u_i. Refer to Fig 5. Consider point p_i and its "neighbors" p_{i-1} and p_{i+1}, for $i = 1, 2, \ldots, h-1$. Consider lines $t(p_{i-1})$ and $t(p_{i+1})$ tangent to C' in p_{i-1} and p_{i+1}, respectively. Further, consider circles C_{i-1} and C_{i+1} centered at p_{i-1} and p_{i+1}, respectively, and passing through p_i. Moreover, consider lines h_{i-1} and h_{i+1} tangent to C_{i-1} and C_{i+1} in p_i, respectively. For each point p_i, with $i = 0, \ldots, h$, consider two half-lines t_1^i and t_2^i incident to p_i, forming angles $\beta_i = \frac{3\alpha}{8h}$ with $t(p_i)$, and both lying in the half-plane delimited by $t(p_i)$ and containing C'. Denote by $W(p_i)$ the wedge delimited by t_1^i and by t_2^i, and containing c.

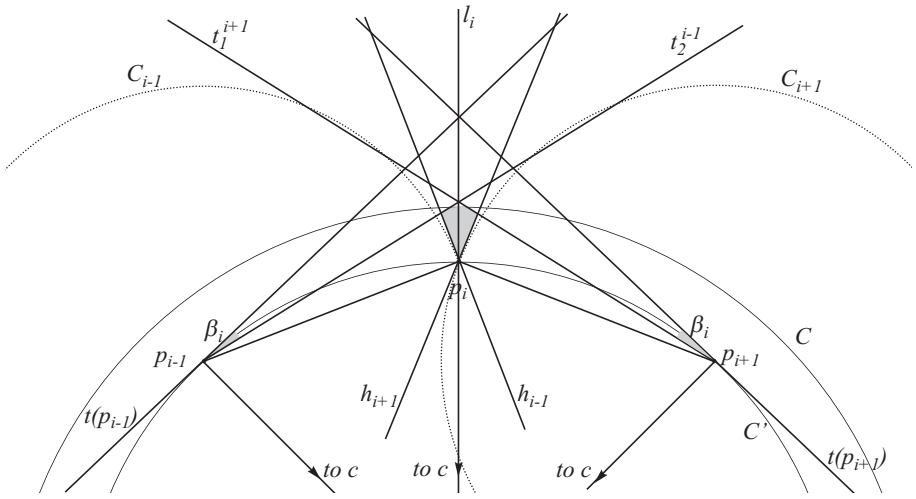


Figure 5: Lines and circles in the construction of Γ. The shaded areas represent angles β_i and region R_i.

We will place Γ_i inside (a part of) the bounded region R_i obtained as the intersection of: (i) the half-plane H^{i-1} delimited by h_{i-1} and not containing C_{i-1}, (ii) the half-plane H^{i+1} delimited by h_{i+1} and not containing C_{i+1}, (iii) wedge $W(p_{i-1})$, (iv) wedge $W(p_{i+1})$, and (v) the disk delimited by C.

First, we prove that R_i is "large enough" to contain Γ_i, namely we claim that there exists an isosceles triangle T that has an angle larger than $\alpha_i = \frac{3\alpha}{16h}$ incident to p_i and that is completely contained in R_i. Such a triangle will have the further feature that the angle incident to p_i is bisected by the line l_i through c and p_i.

Lines h_{i-1} and h_{i+1} are both passing through p_i; we prove that they have different slopes and we compute the angles that they form at p_i. Refer to Fig. 6.

Line h_{i-1} forms an angle of $\frac{\pi}{2}$ with segment $\overline{p_{i-1}p_i}$; angle $\widehat{cp_ip_{i-1}}$ is equal to $\frac{\pi}{2} - \frac{3\alpha}{4h}$, since $\widehat{p_icp_{i-1}} = \frac{3\alpha}{2h}$ and since triangle (p_{i-1}, c, p_i) is isosceles. Hence, the angle delimited by h_{i-1} and l_i is $\pi - \frac{\pi}{2} - (\frac{\pi}{2} - \frac{3\alpha}{4h}) = \frac{3\alpha}{4h}$. Analogously, the angle between l_i and h_{i+1} is $\frac{3\alpha}{4h}$. Hence, the intersection of H^{i-1} and H^{i+1} is a wedge $W(p_i, h_{i-1}, h_{i+1})$ centered at p_i, with an angle of $\frac{3\alpha}{2h}$, and bisected by l_i.

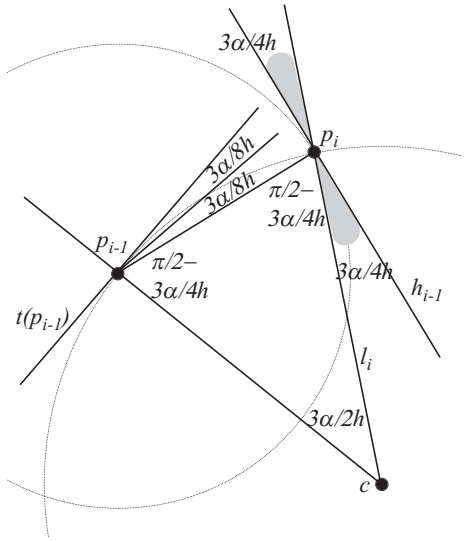


Figure 6: The angle between l_i and h_{i-1}.

We claim that each of t_2^{i-1} and t_1^{i+1} cuts the border of $W(p_i, h_{i-1}, h_{i+1})$ twice. The angle between $t(p_{i-1})$ and $\overline{p_{i-1}p_i}$ is $\frac{3\alpha}{4h}$, because the angle between $t(p_{i-1})$ and $\overline{cp_{i-1}}$ is $\frac{\pi}{2}$, and angle $\widehat{cp_{i-1}p_i}$ is $\frac{\pi}{2} - \frac{3\alpha}{4h}$. The angle between $t(p_{i-1})$ and t_2^{i-1} is $\beta_i = \frac{3\alpha}{8h}$, by construction. Hence, the angle between t_2^{i-1} and $\overline{p_{i-1}p_i}$ is $\frac{3\alpha}{4h} - \frac{3\alpha}{8h} = \frac{3\alpha}{8h}$. Since the slope of both h_{i-1} and h_{i+1} with respect to $\overline{p_{i-1}p_i}$ is greater than $\frac{3\alpha}{8h}$ and smaller than $\pi - \frac{3\alpha}{8h}$, because the slopes of h_{i-1} and h_{i+1} with respect to $\overline{p_{i-1}p_i}$ are $\frac{\pi}{2}$ and $\frac{\pi}{2} - \frac{3\alpha}{4h}$, respectively (notice that $\alpha \leq \frac{\pi}{4}$ and $h \geq 2$), then t_2^{i-1} intersects both h_{i-1} and h_{i+1}. It can be analogously proved that t_1^{i+1} intersects h_{i-1} and h_{i+1}. It follows that the intersection of H^{i-1}, H^{i+1}, $W(p_{i-1})$, and $W(p_{i+1})$ contains a triangle T as required by the claim (notice that the angle of T incident to p_i is $\frac{3\alpha}{2h}$). Considering circle C does not invalidate the existence of T, since C is concentric with C' and has a bigger radius, hence T can always be chosen sufficiently small so that it completely lies inside C.

Now Γ_i can be placed inside T, by scaling Γ_i down till it fits inside T (see Fig. 7.a). The scaling always allows to place Γ_i inside T, since the angle of $R(\Gamma_i)$ incident to p_i is $\alpha_i = \frac{3\alpha}{16h}$, that is smaller than the angle of T incident to p_i, which is $\frac{3\alpha}{2h}$. In particular, we choose to place Γ_i inside T so that l_i bisects the angle of $R(\Gamma_i)$ incident to p_i. This concludes the construction of Γ.

In the following we will prove that the constructed drawing Γ satisfies Prop-

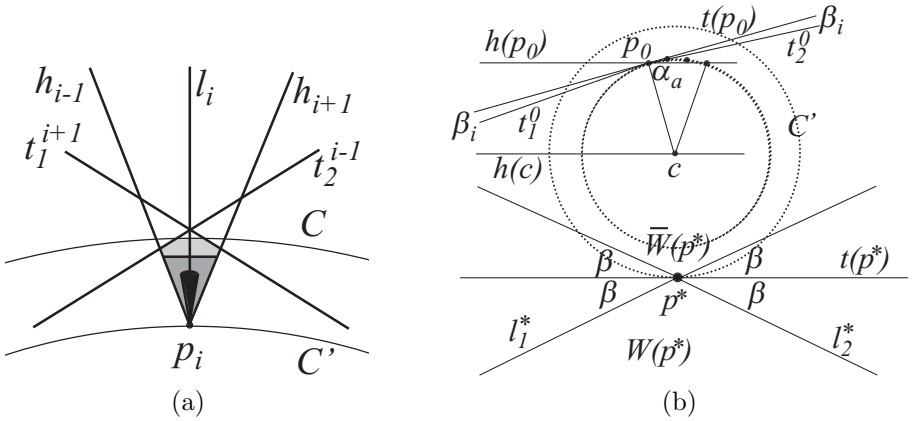


Figure 7: (a) Placement of Γ inside R_i. Region $R(\Gamma)$ is the darkest, triangle T is composed of $R(\Gamma)$ and of the second darkest region, R_i is composed of T and of the light shaded region. (b) Illustration for the proof of Lemma 1.

erties 1–4. However, for this purpose, we need some preliminary lemmata.

Consider the tangent $t(p^*)$ to C in p^*. Consider two half-lines l_1^* and l_2^* incident to p^*, lying in the opposite part of C with respect to $t(p^*)$, and forming angles equal to β with $t(p^*)$. Denote by $W(p^*)$ the wedge centered at p^*, delimited by l_1^* and l_2^*, and not containing C. We have the following lemmata.

Lemma 1 *The closed wedge $W(p^*)$ is completely contained inside the open wedge $W(p_i)$, for each $i = 0, 1, \ldots, h$.*

Proof: Consider any point p_i. First, observe that p_i is contained in the wedge $\overline{W}(p^*)$ obtained by reflecting $W(p^*)$ with respect to $t(p^*)$. Namely, p_i is contained in $W(p^*, \frac{\alpha}{2})$, which is in turn contained inside $\overline{W}(p^*)$, since $\frac{\alpha}{2} < \pi - 2\beta$, as a consequence of the fact that $\frac{\pi}{4} > \beta \geq \alpha$. Hence, in order to prove the lemma, it suffices to show that the absolute value of the slope of each of t_1^i and t_2^i is smaller than the absolute value of the slope of the half-lines delimiting $W(p^*)$. Such latter half-lines form angles of β, by construction, with the x-axis.

The slope of t_1^i can be computed by adding the slope of t_1^i with respect to $t(p_i)$ and the slope of $t(p_i)$. The former slope is equal to $\beta_i = \frac{3\alpha}{8h}$, by construction. Recalling that $t(p_i)$ is the tangent to A in p_i, the slope of $t(p_i)$ is bounded by the maximum among the slopes of the tangents to points of A. Such a maximum is clearly achieved at p_0 and p_h and is equal to $\frac{3\alpha}{4}$. Namely, refer to Fig. 7.b and consider the horizontal lines $h(c)$ and $h(p_0)$ through c and p_0, respectively, that are traversed by radius (c, p_0). Such a radius forms angles of $\frac{\pi}{2}$ with $t(p_0)$; hence, the slope of $t(p_0)$, that is equal to the angle between $t(p_0)$ and $h(p_0)$, is $\frac{\pi}{2}$ minus the angle α_a between $h(p_0)$ and (c, p_0). Angle α_a is the alternate interior of the angle between $h(c)$ and (c, p_0), which is complementary to the half of angle $\widehat{p_0 c p_h}$, which is equal to $\frac{3\alpha}{2}$, by construction. Hence, α_a is equal to $\frac{\pi}{2} - \frac{3\alpha}{4}$ and the slope of $t(p_0)$ is $\frac{3\alpha}{4}$.

It follows that the absolute value of the slope of t_1^i is at most $\frac{3\alpha}{4} + \frac{3\alpha}{8h}$, which is smaller than α, since $h \geq 2$, and hence smaller than β. Analogously, the absolute value of the slope of t_2^i is smaller than β, and the lemma follows. □

Corollary 1 *Point p^* is inside the open wedge $W(p_i)$, for each $i = 1, 2, \ldots, h$.*

Lemma 2 *For every pair of indices i and j such that $1 \leq i < j \leq k$, the drawing of $S(\mu_j)$ is contained inside $W(p_i)$ and the drawing of $S(\mu_i)$ is contained inside $W(p_j)$.*

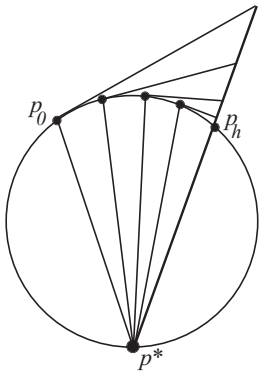


Figure 8: Illustration for the proof of Lemma 2.

Proof: We prove that the drawing of $S(\mu_j)$ is contained inside $W(p_i)$, the proof that the drawing of $S(\mu_i)$ is contained inside $W(p_j)$ being analogous. If $S(\mu_i)$ and $S(\mu_j)$ are consecutive, i.e., the cutvertices parents of $S(\mu_i)$ and $S(\mu_j)$ are u_i and u_j, with $j = i + 1$, then the statement is true by construction. Suppose $S(\mu_i)$ and $S(\mu_j)$ are not consecutive. Refer to Fig. 8. Consider the triangle T_i delimited by (p^*, p_i), by t_2^i, and by the line through p^* and p_h. Such a triangle contains the triangle delimited by (p^*, p_{i+1}), by t_2^{i+1}, and by the line through p^* and p_h, which in turn contains the triangle delimited by (p^*, p_{i+2}), by t_2^{i+2}, and by the line through p^* and p'_b. The repetition of such an argument shows that T_i contains the triangle T_{j-1} delimited by (p^*, p_{j-1}), by t_2^{j-1}, and by the line through p^* and p_h. By construction, Γ_j lies inside T_{j-1}, and the lemma follows. □

We are now ready to prove that the constructed drawing Γ satisfies Properties 1–4.

Property 1. We show that, for every ordered pair of vertices w_1 and w_2, there exists a distance-decreasing path from w_1 to w_2 in Γ. Observe that a distance-decreasing path from w_1 to w_2 is not necessarily a distance-decreasing path from w_2 to w_1. If both w_1 and w_2 are internal to the same graph $S(\mu_i)$, the property follows by induction. If $w_2 = r(\mu)$ and

w_1 is a node in $S(\mu_i)$, then, by Property 3, there exists a path $(w_1 = v_0, v_1, \ldots, v_l = r(\mu_i))$ from w_1 to $r(\mu_i)$ such that, for every point p in $W(p_i)$, $d(v_j, p) < d(v_{j-1}, p)$, for $j = 1, 2, \ldots, l$. By Corollary 1, p^* is contained inside $W(p_i)$. Hence, path $(w_1 = v_0, v_1, \ldots, v_l = r(\mu_i), w_2 = r(\mu))$ is a distance-decreasing path between w_1 and w_2. If $w_1 = r(\mu)$ and w_2 is a node in $S(\mu_i)$, then, by induction, there exists a distance-decreasing path $(v_1 = r(\mu_i), v_2, \ldots, v_l = w_2)$. By Corollary 1, p^* is contained inside $W(p_i)$. Hence, by Property 4, $d(p_i, w_2) < d(p^*, w_2)$. It follows that path $(w_1 = r(\mu), v_1 = r(\mu_i), v_2, \ldots, v_l = w_2)$ is a distance-decreasing path between w_1 and w_2. If w_1 belongs to $S(\mu_i)$ and w_2 belongs to $S(\mu_k)$ then suppose, w.l.o.g., that $k > i$. We show the existence of a distance-decreasing path $\mathcal{P}$ in Γ, composed of three subpaths $\mathcal{P}_1, \mathcal{P}_2$, and $\mathcal{P}_3$. By Property 3, Γ_i is such that there exists a path $\mathcal{P}_1 = (w_1 = v_0, v_1, \ldots, v_l = r(\mu_i))$ from w_1 to $r(\mu_i)$ such that, for every point p in $W(p_i)$, $d(v_j, p) < d(v_{j-1}, p)$, for $j = 1, 2, \ldots, l$. By Lemma 2, drawing Γ_k, and hence vertex w_2, is contained inside $W(p_i)$. Hence, at every vertex of path $\mathcal{P}_1$, the distance from w_2 decreases. Path $\mathcal{P}_2 = (u_i = r(\mu_i), u_{i+1}, \ldots, u_k = r(\mu_k))$ is easily shown to be distance-decreasing with respect to w_2. In fact, for each $l = i, i+1, \ldots, k-2$, angle $u_l\widehat{u_{l+1}u_k}$ is greater than $\frac{\pi}{2}$, because triangle (u_l, u_{l+1}, u_k) is inscribed in less than half a circle with u_{l+1} as middle point. Angle $u_l\widehat{u_{l+1}w_2}$ is strictly greater than $u_l\widehat{u_{l+1}u_k}$, hence it is the biggest angle in triangle (u_l, u_{l+1}, w_2), which implies $d(u_{l+1}, w_2) < d(u_l, w_2)$. By induction, there exists a distance-decreasing path $\mathcal{P}_3$ from $r(\mu_k)$ to w_2, thus obtaining a distance-decreasing path $\mathcal{P}$ from w_1 to w_2.

Property 2. Such a property holds for Γ by construction.

Property 3. Consider any node v in $S(\mu_i)$ and consider any point p internal to $W(p^*)$. By Lemma 1, p is internal to $W(p_i)$, as well. By induction, there exists a path $(v = v_0, v_1, \ldots, v_l = r(\mu_i))$ such that $d(v_j, p) < d(v_{j-1}, p)$, for $j = 1, 2, \ldots, l$. Hence, path $(v = v_0, v_1, \ldots, v_l = r(\mu_i), v_{l+1} = r(\mu))$ is a path such that $d(v_j, p) < d(v_{j-1}, p)$, for $j = 1, 2, \ldots, l+1$, if and only if $d(r(\mu), p) < d(r(\mu_i), p)$. Angle $p\widehat{p^*r(\mu_i)}$ is at least $\beta + (\frac{\pi}{2} - \frac{\alpha}{2})$, which is greater than $\frac{\pi}{2}$. It follows that $(p, r(\mu_i))$ is the longest side of triangle $(p, p^*, r(\mu_i))$, thus proving that $d(p, p^*) < d(p, r(\mu_i))$ and that Property 3 holds for Γ.

Property 4. By Property 2, v is contained inside the wedge $W(p^*, \alpha)$ with angle α, centered at p^*, and bisected by the line through p^* and c. Consider any point p inside $W(p^*)$. Angle $p\widehat{p^*v}$ is at least $\beta + (\frac{\pi}{2} - \frac{\alpha}{2})$, which is greater than $\frac{\pi}{2}$. It follows that (p, v) is the longest side of triangle (p, p^*, v), thus proving that $d(p, v) < d(p^*, v)$ and that Property 4 holds for Γ.

When the induction is performed with μ equal to the root ν of the BC-tree $\mathcal{T}$, we obtain a greedy drawing of S, thus proving the following:

Theorem 2 *There exists an algorithm that constructs a greedy drawing of any triangulated binary cactus.*

4 Spanning a Triangulation with a Binary Cactus

In this section we prove the following theorem:

Theorem 3 *Given a triangulation G, there exists a spanning subgraph S of G such that S is a triangulated binary cactus.*

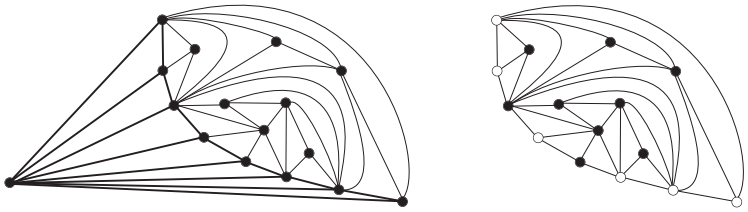
Consider any triangulation G. We are going to construct a binary cactus S spanning G. First, we outline the algorithm to construct S. Such an algorithm has several steps. At the first step, we choose a vertex u incident to the outer face of G and we construct a triangulated cycle C_T composed of u and of all its neighbors. We remove u and its incident edges from G, obtaining a biconnected internally-triangulated plane graph G^*. At the beginning of each step after the first one, we suppose to have already constructed a binary cactus S whose vertices are a subset of the vertices of G (at the beginning of the second step, S coincides with C_T), and we assume to have a set $\mathcal{G}$ of subgraphs of G (at the beginning of the second step, G^* is the only graph in $\mathcal{G}$). Each of such subgraphs is biconnected, internally-triangulated, has an outer face whose vertices already belong to S, and has internal vertices. All such internal vertices do not belong to S and each vertex of G not belonging to S is internal to a graph in $\mathcal{G}$. Only one of the graphs in $\mathcal{G}$ may have chords. During each step, we perform the following two actions:

- *Action 1.* We partition the only graph G_C of $\mathcal{G}$ with chords, if any, into several biconnected internally-triangulated chordless plane graphs; we remove G_C from $\mathcal{G}$ and we add to $\mathcal{G}$ all the graphs with internal vertices into which G_C has been partitioned.
- *Action 2.* We choose a graph G_i from $\mathcal{G}$, we choose a vertex u incident to the outer face of G_i and already belonging to exactly one block of S, and we add to S a block composed of u and of all its neighbors internal to G_i. We remove u and its incident edges from G_i, obtaining a biconnected internally-triangulated plane graph G_i^*. We remove G_i from $\mathcal{G}$ and we add G_i^* to $\mathcal{G}$.

The algorithm stops when $\mathcal{G}$ is empty, that is, when all the vertices of G are spanned by S. An exemplary execution of the algorithm is shown in Figs. 9–16.

Now we give the details of the above outlined algorithm. At the first step of the algorithm, choose any vertex u incident to the outer face of G. Consider all the neighbors $(u_1, u_2, \ldots, u_l)$ of u in clockwise order around it. Since G is a triangulation, $C = (u, u_1, u_2, \ldots, u_l)$ is a cycle, hence the subgraph of G composed of C and of the edges connecting u to its neighbors is a triangulated cycle C_T. Let $S = C_T$. Remove vertex u and all its incident edges from G, obtaining a biconnected internally-triangulated plane graph G^*.

If G^* has no internal vertex, then all the vertices of G belong to S and we have the desired binary cactus spanning G. Otherwise, G^* has internal vertices. Let $\mathcal{G} = \{G^*\}$.



(a) (b)

Figure 9: First step of the algorithm: (a) A triangulation G, from which a vertex u and its neighbors are selected. The thick subgraph is the triangulated cycle C_T such that $S = C_T$ after Step 1. (b) Graph G^* obtained from G by removing u and its incident edges. Two arbitrarily chosen vertices (represented by black circles) incident to $f(G^*)$ are forbidden for G^*, all others (represented by white circles) are assigned to it.

At each step of the algorithm, for each graph $G_i \in \mathcal{G}$, consider the vertices incident to $f(G_i)$. Each of such vertices can be either *forbidden for* G_i or *assigned to* G_i. A vertex w is forbidden for G_i if the following choice has been done: S will contain no block incident to w and spanning a subgraph of G_i. Conversely, a vertex w is assigned to G_i if a new block incident to w and spanning a subgraph of G_i could be introduced in S. For example, w is forbidden for G_i if two blocks of S already exist sharing w as a cutvertex. At the end of the first step of the algorithm, choose any two vertices incident to $f(G^*)$ as the only forbidden vertices for G^*. All the other vertices incident to $f(G^*)$ are assigned to G^*.

At the beginning of the i-th step, with $i \geq 2$, we assume that each of the following holds:

- *Invariant A:* Graph S is a binary cactus spanning all and only the vertices that are not internal to any graph in $\mathcal{G}$.
- *Invariant B:* Each graph in $\mathcal{G}$ is biconnected, internally-triangulated, and has internal vertices.
- *Invariant C:* Only one of the graphs in $\mathcal{G}$ may have chords.
- *Invariant D:* No internal vertex of a graph $G_i \in \mathcal{G}$ belongs to a graph $G_j \in \mathcal{G}$, with $i \neq j$.
- *Invariant E:* For each graph $G_i \in \mathcal{G}$, all the vertices incident to $f(G_i)$ are assigned to G_i, except for two vertices, which are forbidden.
- *Invariant F:* Each vertex v incident to the outer face of a graph in $\mathcal{G}$ is assigned to at most one graph $G_i \in \mathcal{G}$. If a vertex v incident to the outer face of a graph in $\mathcal{G}$ is assigned to a graph $G_i \in \mathcal{G}$, then v is forbidden for all graphs $G_j \in \mathcal{G}$ such that v is incident to $f(G_j)$, with $j \neq i$.

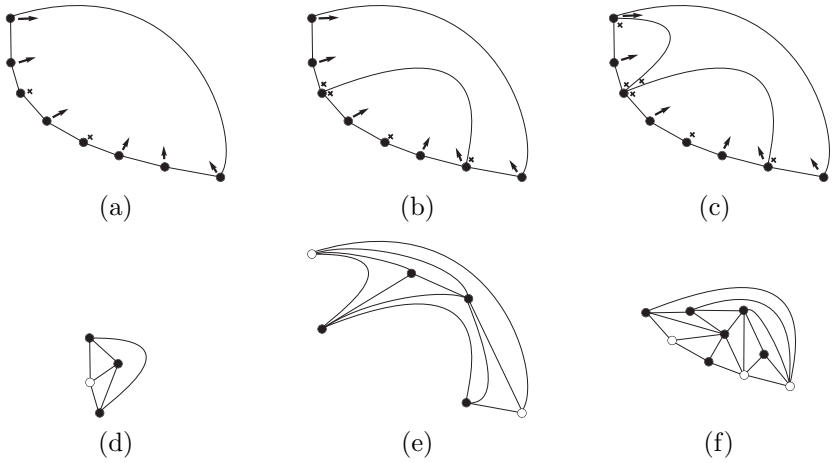


Figure 10: Step 2, Action 1. (a)–(c) Outerplane graphs O_C^0, O_C^1, and $O_C^2 = O_C$, and the assignment of vertices to their faces. (d)–(f) Graphs G_1, G_2, and G_3, where $\mathcal{G} = \{G_1, G_2, G_3\}$, obtained by partitioning G^* into biconnected, internally triangulated, chordless subgraphs.

- *Invariant G:* Each vertex assigned to a graph in $\mathcal{G}$ belongs to exactly one block of S.

Such invariants clearly hold after the first step of the algorithm. During each step of the algorithm after the first one, we perform the following two actions.

Action 1: If $\mathcal{G}$ does not contain any graph with chords, go to Action 2. Otherwise, by Invariant C, only one of the graphs in $\mathcal{G}$, say G_C, has chords. We use the chords of G_C to partition it into k biconnected, internally-triangulated, chordless graphs G_C^j, with $j = 1, 2, \ldots, k$.

Consider the subgraph O_C of G_C induced by the vertices incident to $f(G_C)$. Clearly, O_C is a biconnected outerplane graph. To each internal face f of O_C

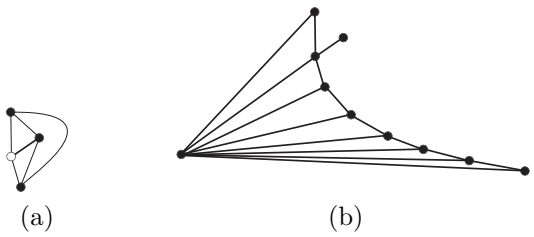
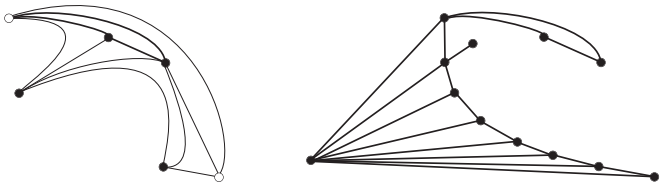
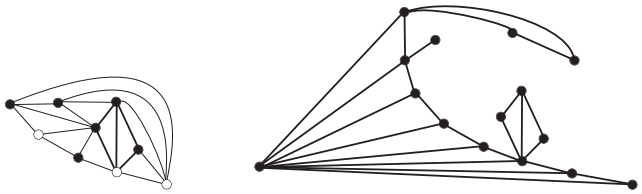


Figure 11: Step 2, Action 2. (a) Choice of a graph G_i in $\mathcal{G}$ (here $G_i = G_1$) and of a vertex u incident to $f(G_i)$. The thick subgraph is the edge (u, u_1) added to S after Action 2 of Step 2. (b) Binary cactus S after Action 2 of Step 2. Set $\mathcal{G}$ is now $\{G_2, G_3\}$.



(a) (b)

Figure 12: Step 3, Action 2 (Action 1 of Step 3 is skipped because no graph in $\mathcal{G}$ has chords). (a) Choice of a graph G_i in $\mathcal{G}$ (here $G_i = G_2$) and of a vertex u incident to $f(G_i)$. The thick subgraph is the triangulated cycle C_T added to S after Step 3, Action 2. (b) Binary cactus after Action 2 of Step 3. Set $\mathcal{G}$ is now $\{G_3\}$.



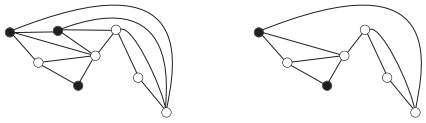
(a) (b)

Figure 13: Step 4, Action 2 (Action 1 of Step 4 is skipped because no graph in $\mathcal{G}$ has chords). (a) Choice of a graph G_i in $\mathcal{G}$ (here $G_i = G_3$) and of a vertex u incident to $f(G_i)$. The thick subgraph is the triangulated cycle C_T added to S after Step 4, Action 2. (b) Binary cactus S after Action 2 of Step 4. Set $\mathcal{G}$ is now $\{G_3^*\}$, where G_3^* is the graph obtained from G_3 by removing u and its incident edges.

delimited by a cycle C, a graph G_C^j is associated such that G_C^j is the subgraph of G_C induced by the vertices of C or inside C. We are going to replace G_C with graphs G_C^j in $\mathcal{G}$. However, we first show how to decide which vertices incident to the outer face of a graph G_C^j are assigned to G_C^j and which vertices are forbidden for G_C^j. Since each graph G_C^j is univocally associated with a face of O_C, in the following we assign vertices to the faces of O_C and we forbid vertices for the faces of O_C, meaning that if a vertex is assigned to (forbidden for) a face f of O_C, then it is assigned to (resp. forbidden for) the associated graph G_C^j.

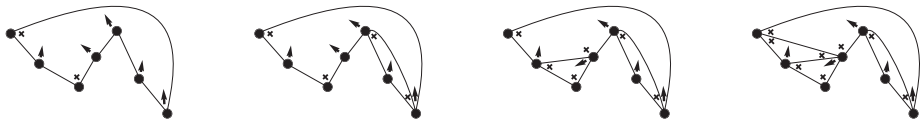
We want to assign the vertices incident to $f(O_C)$ to faces of O_C so that:

- Property 1: No forbidden vertex is assigned to any face of O_C;
- Property 2: No vertex is assigned to more than one face of O_C;
- Property 3: Each face of O_C has exactly two incident vertices which are forbidden for it; all the other vertices of the face are assigned to it.



(a) (b)

Figure 14: Step 5, before Action 1. (a) The only graph $G_C = G_3^*$ in $\mathcal{G}$, with its assigned vertices (white circles) and forbidden vertices (black circles). (b) The outerplane graph O_C induced by the vertices incident to $f(G_C)$.



(a) (b) (c) (d)

Figure 15: Step 5, Action 1. (a)–(d) Outerplane graphs O_C^0, O_C^1, O_C^2, $O_C^3 = O_C$, and the assignment of vertices to their faces. Partitioning G_C into subgraphs G_C^j produces only one graph, say G_4, with internal vertices. Hence, set $\mathcal{G}$ is now $\{G_4\}$.

By Invariant E, G_C has two forbidden vertices. We construct an assignment of vertices to the faces of O_C in some steps. Let p be the number of chords of O_C. Consider the Hamiltonian cycle O_C^0 of O_C, and assign all the vertices of O_C^0, but for the two forbidden vertices, to the only internal face of O_C^0. At the i-th step, $1 \leq i \leq p$, we insert into O_C^{i-1} a chord of O_C, obtaining a graph O_C^i. This is done so that Properties 1–3 are satisfied by O_C^i (with O_C^i instead of O_C). After all the p chords of O_C have been inserted, $O_C^p = O_C$, and we have an assignment of vertices to faces of O_C satisfying Properties 1–3.

Properties 1–3 are clearly satisfied by the assignment of vertices to the faces of O_C^0. Inductively assume that Properties 1–3 are satisfied by the assignment of vertices to the faces of O_C^{i-1}. Let (u_a, u_b) be the chord that is inserted at the i-th step. Chord (u_a, u_b) partitions a face f of O_C^{i-1} into two faces f_1 and f_2. By Property 3, two vertices u_1^* and u_2^* incident to f are forbidden for it and all other vertices incident to f are assigned to it. For each face of O_C^i different from f_1 and f_2, assign and forbid vertices as in the same face in O_C^{i-1}. Assign and forbid vertices for f_1 and f_2 as follows:

- If vertices u_a and u_b are the same vertices as u_1^* and u_2^* (see Fig. 17), assign to f_1 and f_2 all the vertices incident to it, except for u_a and u_b. No forbidden vertex has been assigned to any face of O_C^i (Property 1). Vertices u_a and u_b have not been assigned to any face. All the vertices assigned to f belong to exactly one of f_1 and f_2 and so they have been assigned to exactly one face (Property 2). The only vertices of f_1 (resp. of f_2) not assigned to it are u_a and u_b, while all the other vertices are assigned to such a face (Property 3).

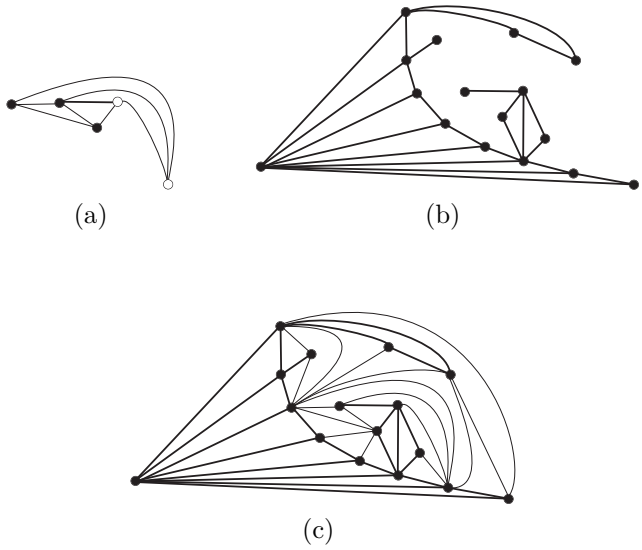


Figure 16: Step 5, Action 2: (a) Choice of a graph G_i in $\mathcal{G}$ (here $G_i = G_4$) and of a vertex u incident to $f(G_i)$. The thick subgraph is the edge (u, u_1) added to S after Step 5, Action 2. (b) Binary cactus S at the end of the algorithm. (c) The obtained binary cactus S spans G.

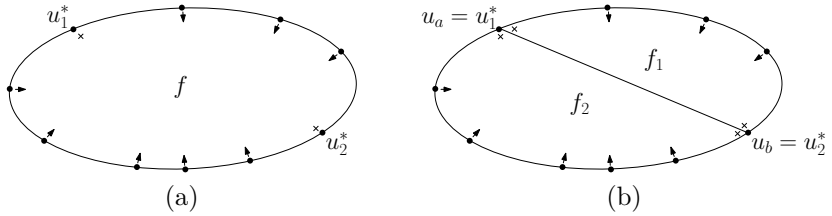


Figure 17: Vertices u_a and u_b are the same vertices as u_1^* and u_2^*.

- If vertices u_a and u_b are both distinct from each of u_1^* and u_2^*, and u_1^* and u_2^* are either both in f_1 or both in f_2, say in f_1 (see Fig. 18), assign to f_1 all the vertices incident to it, except for u_1^* and u_2^*, and assign to f_2 all the vertices incident to it, except for u_a and u_b. No forbidden vertex has been assigned to any face of O_C^i (Property 1). Vertices u_a and u_b have been assigned to exactly one face, namely f_1. All the other vertices assigned to f belong to exactly one of f_1 or f_2 and so they have been assigned to exactly one face (Property 2). The only vertices of f_1 (resp. of f_2) not assigned to it are u_1^* and u_2^* (resp. u_a and u_b), while all the other vertices are assigned to such a face (Property 3).

- If vertices u_a and u_b are both distinct from each of u_1^* and u_2^* and one of u_1^* and u_2^*, say u_1^*, is in f_1 while u_2^* is in f_2 (see Fig. 19), assign to f_1 all

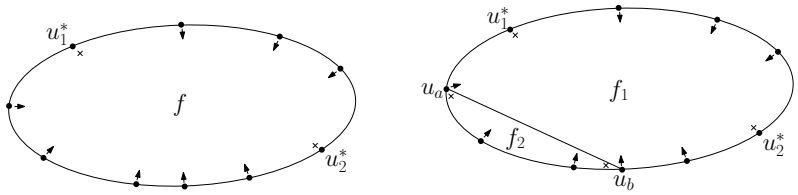


Figure 18: Vertices u_a and u_b are both distinct from each of u_1^* and u_2^* and both u_1^* and u_2^* are in f_1.

the vertices incident to it, except for u_1^* and u_a, and assign to f_2 all the vertices incident to it, except for u_2^* and u_b. No forbidden vertex has been assigned to any face of O_C^i (Property 1). Vertices u_a and u_b have been assigned to exactly one face, namely f_2 and f_1, respectively. All the other vertices assigned to f belong to exactly one of f_1 and f_2 and so they have been assigned to exactly one face (Property 2). The only vertices of f_1 (resp. of f_2) not assigned to it are u_1^* and u_a (resp. u_2^* and u_b), while all the other vertices are assigned to such a face (Property 3).

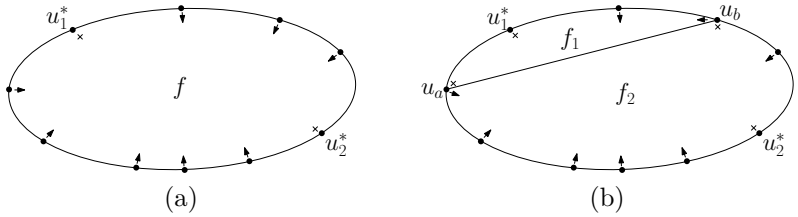


Figure 19: Vertices u_a and u_b are both distinct from each of u_1^* and u_2^*, u_1^* is in f_1, and u_2^* is in f_2.

- If one of the vertices u_1^* and u_2^* coincides with one of u_a and u_b, say u_1^* coincides with u_a, and u_2^* is in one of f_1 and f_2, say in f_1 (see Fig. 20), assign to f_1 all the vertices incident to it, except for u_2^* and u_a, and assign to f_2 all the vertices incident to it, except for u_a and u_b. No forbidden vertex has been assigned to any face of O_C^i (Property 1). Vertex u_a has not been assigned to any face and vertex u_b has been assigned to exactly one face, namely f_1. All the other vertices assigned to f belong to exactly one of f_1 and f_2 and so they have been assigned to exactly one face (Property 2). The only vertices of f_1 (resp. of f_2) not assigned to it are u_2^* and u_a (resp. u_a and u_b), while all the other vertices are assigned to such a face (Property 3).

Graph G_C is removed from $\mathcal{G}$. All the graphs G_C^j having internal vertices are added to $\mathcal{G}$. We prove that Invariants A–G are satisfied after Action 1.

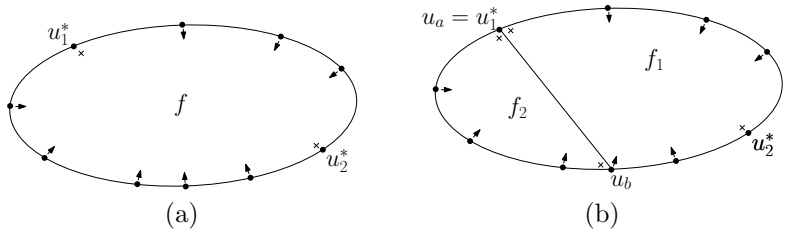


Figure 20: Vertex u_1^* coincides with u_a and vertex u_2^* is in f_1.

Invariant A: A vertex is internal to a graph in $\mathcal{G}$ after Action 1 if and only if it is internal to a graph in $\mathcal{G}$ before Action 1. Since no block is added to S during Action 1, then Invariant A holds after Action 1.

Invariant B: By construction, each graph G_C^j inserted into $\mathcal{G}$ after Action 1 has internal vertices. Further, G_C^j is the graph contained inside a simple cycle of a biconnected internally triangulated plane graph, hence it is biconnected and internally triangulated, as well, satisfying Invariant B.

Invariant C: By Invariant C, before Action 1 only graph G_C may have chords among the graphs in $\mathcal{G}$. After Action 1, however, G_C is replaced in $\mathcal{G}$ by chordless graphs and hence no graph in $\mathcal{G}$ has chords, satisfying Invariant C.

Invariant D: By Invariant D, each vertex that, before Action 1, is internal to a graph $G_i \neq G_C$ in $\mathcal{G}$ does not belong to any graph $G_j \neq G_i$ in $\mathcal{G}$. Since the set of vertices belonging to graphs G_C^j is a subset of the vertices of G_C, after Action 1 Invariant D holds for all the vertices internal to a graph $G_i \neq G_C^j$. An internal vertex of a graph G_C^j is an internal vertex of G_C, as well, hence, by Invariant D, it does not belong to any graph that has not been introduced in $\mathcal{G}$ during Action 1. It remains to prove that an internal vertex of a graph G_C^j does not belong to any graph G_C^l, with $l \neq j$. By construction, the internal vertices of such graphs are inside cycles corresponding to distinct faces of O_C. Hence, an internal vertex of G_C^j does not belong to G_C^l.

Invariant E: Invariant E holds for all the graphs that are in $\mathcal{G}$ before Action 1 and that are still in $\mathcal{G}$ after Action 1. By Property 3, each graph G_C^j inserted into $\mathcal{G}$ after Action 1 satisfies Invariant E.

Invariant F: All the vertices that, before Action 1, are assigned to a graph $G_i \neq G_C$ in $\mathcal{G}$ satisfy Invariant F after Action 1. Namely, by Invariant F before Action 1, if they are incident to $f(G_C)$, then they are forbidden for G_C and, by Property 1, they are not assigned to any graph G_C^j. By Invariant F, before Action 1 each vertex w assigned to G_C is not assigned to any graph $G_i \neq G_C$ in $\mathcal{G}$. After Action 1, G_C is not a graph in $\mathcal{G}$ any longer, hence w is not assigned to it. By Property 2, after Action 1 each

vertex is assigned to at most one graph G_C^j, hence Invariant F holds after Action 1.

Invariant G: Since no block is added to S during Action 1, and since the set of vertices assigned to graphs in $\mathcal{G}$ after Action 1 is a subset of the set of vertices assigned to graphs in $\mathcal{G}$ before Action 1, then Invariant G holds after Action 1.

Action 2: After Action 1 all graphs in $\mathcal{G}$ are chordless. Notice that there is at least one graph G_i in $\mathcal{G}$, otherwise the algorithm would have stopped before Action 1. By Invariant B, G_i has internal vertices. Choose any vertex u that is incident to $f(G_i)$ and that is assigned to G_i (see Fig. 21). By the biconnectivity of G_i and by the fact that it has internal vertices, $f(G_i)$ has at least three vertices. Since each graph in $\mathcal{G}$ has at most two forbidden vertices (by Invariant E), a vertex u assigned to G_i always exists. Consider all the neighbors $(u_1, u_2, \ldots, u_l)$ of u internal to G_i, in clockwise order around u. Since G is biconnected, chordless, internally triangulated, and has internal vertices, then $l \geq 1$. If $l = 1$, then let C_T be edge (u, u_1). Otherwise, let C_T be the triangulated cycle composed of cycle $(u, u_1, u_2, \ldots, u_l)$ and of the edges connecting u to its neighbors. Add C_T to S. Remove u and its incident edges from G_i, obtaining a graph G_i^*. Assign to G_i^* all the vertices incident to $f(G_i^*)$, except for the two vertices that are forbidden for G_i. Remove G_i from $\mathcal{G}$ and insert G_i^*, if it has internal vertices, into $\mathcal{G}$.

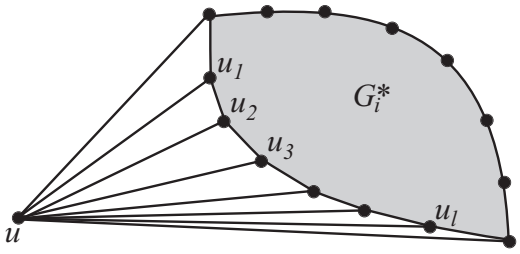


Figure 21: Action 2 of a step of the algorithm.

We prove that Invariants A–G are satisfied after Action 2.

Invariant A: The block $(u, u_1, u_2, \ldots, u_l)$ added to S is either an edge or a triangulated cycle. By Invariant A, before Action 2 all vertices internal to a graph in $\mathcal{G}$ are not spanned by S. Further, by Invariant G, before Action 2 vertex u belongs to exactly one block of S. It follows that S is still a binary cactus after Action 2. Before Action 2, S spans all and only the vertices that are not internal to any graph in $\mathcal{G}$. The only vertices that are internal to a graph in $\mathcal{G}$ before Action 2 and that are incident to the outer face of a graph in $\mathcal{G}$ after Action 2 are $u_1, u_2, \ldots, u_l$, which are spanned by S after Action 2. Hence, S spans all the vertices of G that are not internal to any graph in $\mathcal{G}$. Before Action 2, no internal vertex of

a graph in G is spanned by S. The vertices which are added to S during Action 2 are incident to $f(G_i^*)$, hence, by Invariant D to be proved below, they are not internal to any graph in $\mathcal{G}$ after Action 2. Hence, S does not span vertices of G that are internal to a graph in $\mathcal{G}$, satisfying Invariant A.

Invariant B: By construction, G_i^* is the only graph inserted into $\mathcal{G}$ after Action 2. However, G_i^* is biconnected and internally triangulated, since it is obtained from a graph G_i that, by Invariant B before Action 2, is biconnected, internally triangulated, chordless, and has internal vertices, by removing a vertex incident to $f(G_i)$. Further, G_i^* has internal vertices, otherwise it would not have been inserted into $\mathcal{G}$. Hence, Invariant B is satisfied after Action 2.

Invariant C: Before Action 2, all graphs in $\mathcal{G}$ have no chord. At most one graph, namely G_i^*, is inserted into $\mathcal{G}$ after Action 2, hence Invariant C is still satisfied.

Invariant D: By Invariant D, before Action 2 no internal vertex of a graph $G_l \neq G_i$ in $\mathcal{G}$ belongs to a graph $G_j \neq G_l$ in $\mathcal{G}$. Since the vertices of G_i^* are a subset of the vertices of G_i then, after Action 2, Invariant D holds for each internal vertex of G_l. Further, the internal vertices of G_i^* are a subset of the internal vertices of G_i and hence, after Action 2, Invariant D holds also for each internal vertex of G_i^*.

Invariant E: Invariant E holds for all the graphs that are in $\mathcal{G}$ before Action 2 and that are still in $\mathcal{G}$ after Action 2. By construction, all the vertices incident to the outer face of G_i^*, except for the two forbidden vertices of G_i, are assigned to G_i^*, satisfying Invariant E.

Invariant F: The only vertices that are assigned to a graph in $\mathcal{G}$ during Action 2 are the vertices incident to the outer face of G_i^*. All the vertices internal to G_i before Action 2 and incident to the outer face of G_i^* after Action 2 are assigned exclusively to G_i^*, namely if before Action 2 one of such vertices is assigned to a graph $G_j \neq G_i$, then such a vertex would be incident to the outer face of G_j, contradicting Invariant D. All the vertices that are assigned to G_i before Action 2 and that are incident to the outer face of G_i^* after Action 2, are assigned exclusively to G_i before Action 2, by Invariant F, and hence they are assigned only to G_i^* after Action 2. All the vertices that are assigned to a graph different from G_i are such that, if they are incident to the outer face of G_i, then they are forbidden for it. Since all the vertices forbidden for G_i are forbidden for G_i^*, then Invariant F holds for such vertices, as well.

Invariant G: The block added to S after Action 2 spans only vertices internal to G_i and vertex u. Hence, all the vertices assigned to a graph in $\mathcal{G}$ and not belonging to G_i are still spanned by a single block of S. All the vertices incident to the outer face of G_i, except for u, are not spanned by the

block added during Action 2. All the vertices internal to G_i and assigned to G_i^* are spanned by the only block added during Action 2. Finally, after Action 2, vertex u is not assigned to any graph in $\mathcal{G}$ any longer.

When the algorithm stops, i.e., when there is no graph in $\mathcal{G}$, by Invariant A graph S is a binary cactus spanning all vertices of G, hence proving Theorem 3.

5 Extension to Triconnected Planar Graphs and Conclusions

In this paper we have shown an algorithm for constructing greedy drawings of triangulations. The algorithm relies on two main results. The first one states that every triangulated binary cactus admits a greedy drawing. The second one states that, for every triangulation G, there exists a triangulated binary cactus S spanning G.

After the conference version of this paper was submitted, we realized that slight modifications of the two main arguments, presented in Sect. 3 and Sect. 4, prove Conjecture 1. In the following we deal with such a result.

First, observe that, given a triangulated binary cactus S spanning a triangulation G, the internal edges of each triangulated cycle of S could be removed, still leaving S a spanning subgraph of G. Hence, every triangulation can be spanned by a *non-triangulated binary cactus*, which is a connected graph such that: (i) the block associated with each B-node of $\mathcal{T}$ is either an edge or a simple cycle; and (ii) every cutvertex is shared by exactly two blocks of S.

Second, it is not difficult to argue that the algorithm shown in Sect. 3 also constructs greedy drawings of any non-triangulated binary cactus S. More specifically, construct the BC-tree $\mathcal{T}$ of S; consider each block $(r(\mu) = u_0, u_1, \ldots, u_{h-1})$ corresponding to a B-node μ of $\mathcal{T}$ and insert a dummy edge between $r(\mu)$ and each node u_i, with $1 \leq i \leq h-2$; the resulting graph S' is a triangulated binary cactus; apply the algorithm described in Sect. 3 to construct a greedy drawing Γ' of S'; finally, remove dummy edges from Γ', obtaining a drawing Γ of S.

We claim that Γ is a greedy drawing. Notice that the validity of Lemmata 1 and 2 only depends on the angles of the geometric construction. Hence, such Lemmata hold for Γ. Then, it is enough to prove that at each step of the induction Γ satisfies Properties 1–4 described in Sect. 3.

Actually, Property 2 and Property 4 are trivially verified, since they only depend on the angles of the construction.

The proof of Property 1 can be conducted analogously to the one presented in Sect. 3, namely by proving that, for every pair of vertices w_1 and w_2, there exists a distance-decreasing path between them. However, the case in which the distance-decreasing path contains edge $(u_i, r(\mu))$, for some $2 \leq i \leq h-2$, deserves an explicit discussion, because such an edge is no longer an edge of the graph. Observe that it can be supposed that one out of w_1 and w_2 is $r(\mu)$,

because in all the other cases the distance-decreasing path between w_1 and w_2 does not contain $(u_i, r(\mu))$.

First, suppose that the path ends at $r(\mu)$, i.e., $w_2 = r(\mu)$. Edge $(u_i, r(\mu))$ can be replaced either by path $(u_i, u_{i-1}, \ldots, u_1, u_0)$ or by path $(u_i, u_{i+1}, \ldots, u_{h-1}, u_0)$, depending on whether $i \leq \frac{h}{2}$ or $i \geq \frac{h}{2}$, still leaving the path distance-decreasing. In fact (see Fig.22.a), denote by p' the intersection point between C' and segment $\overline{cp^*}$ and suppose that $i \geq \frac{h}{2}$, the case in which $i \leq \frac{h}{2}$ being analogous; angle $\widehat{u_i u_{i+1} p'}$ is greater than or equal to $\frac{\pi}{2}$ because triangle (u_i, u_{i+1}, p') is inscribed in no more than half a circle with u_{i+1} as middle point; then, angle $\widehat{u_i u_{i+1} p^*}$ is also greater than $\frac{\pi}{2}$ because it is strictly greater than $\widehat{u_i u_{i+1} p'}$; hence, $\overline{p^* u_i}$ is longer than $\overline{p^* u_{i+1}}$; it follows that, when traversing edge (u_i, u_{i+1}), the path decreases its distance from the point p^* where $r(\mu)$ is drawn.

Second, suppose that the path starts at $r(\mu)$, i.e., $w_1 = r(\mu)$. Edge $(r(\mu), u_i)$ can be replaced either by path $(u_0, u_1, \ldots, u_{i-1}, u_i)$ or by path $(u_0, u_{h-1}, \ldots, u_{i+1}, u_i)$, depending on whether $i \leq \frac{h}{2}$ or $i \geq \frac{h}{2}$, still leaving the path distance-decreasing. In fact, suppose that $i \geq \frac{h}{2}$, the case in which $i \leq \frac{h}{2}$ being analogous; as in the previous case, edge $(r(\mu), u_{h-1})$ can be shown to decrease the distance from w_2 by considering triangle $(r(\mu), u_{h-1}, w_2)$ and arguing that angle $\widehat{p^* u_{h-1} w_2}$ is greater than $\frac{\pi}{2}$. Further, path $(u_{h-1}, u_{h-2}, \ldots, u_{i+1}, u_i, \ldots, w_2)$ can be shown to be distance-decreasing as in the proof of Property 1 in Sect. 3 (in the case in which w_1 belongs to $S(\mu_i)$ and w_2 belongs to $S(\mu_j)$).

In order to prove Property 3, it is sufficient to observe that an edge $(u_i, r(\mu))$ can be replaced either by path $(u_i, u_{i-1}, \ldots, u_1, u_0)$ or by path $(u_i, u_{i+1}, \ldots, u_{h-1}, u_0)$, still obtaining a path in which at every step the distance from any point in $W(p^*)$ decreases. In fact (see Fig.22.b), denote by p any point inside $W(p^*)$, and denote by $a_{i-1,i}$ and $a_{i,i+1}$ the axes of segments $\overline{p_{i-1} p_i}$ and $\overline{p_i p_{i+1}}$, respectively. Since $a_{i-1,i}$ and $a_{i,i+1}$ intersect in the center of C', we have that p is either to the left of $a_{i-1,i}$ or to the right of $a_{i,i+1}$, or both. Suppose that p is to the right of $a_{i,i+1}$, the other case being analogous. Then, $d(p, p_{i+1}) < d(p, p_i)$. The repetition of such an argument leads to prove that path $(u_i, u_{i+1}, \ldots, u_{h-1}, u_0)$ decreases the distance from p at every vertex.

As we proved that there exists an algorithm to construct greedy drawings of non-triangulated binary cactuses, in order to prove Conjecture 1 it suffices to show that every triconnected planar graph admits a non-triangulated binary cactus as a spanning subgraph. In the following we sketch how to extend the arguments of Sect. 4 in order to prove such a result.

The algorithm to find a non-triangulated binary cactus spanning a given triconnected planar graph G consists of several steps, in which the cactus is constructed incrementally by adding to it one block at a time. As in the triangulated case, at the beginning of each step after the first one, we suppose to have already constructed a non-triangulated binary cactus S whose vertices are a subset of the vertices of G, and we assume to have a set $\mathcal{G}$ of subgraphs of G. Further, we assume that the following invariants hold:

- *Invariant A:* Graph S is a non-triangulated binary cactus spanning all

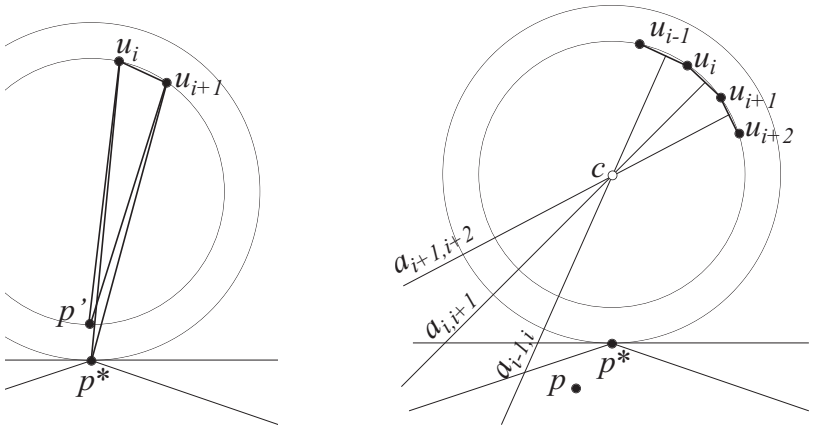


Figure 22: (a) When traversing edge (u_i, u_{i+1}), the distance from p^* decreases. (b) When traversing edge (u_i, u_{i+1}), the distance from p decreases.

and only the vertices that are not internal to any graph in $\mathcal{G}$.

- *Invariant B:* Each graph in $\mathcal{G}$ is biconnected and has internal vertices.
- *Invariant C:* At most one graph $G_C \in \mathcal{G}$ has separation pairs. However, if G_C exists, each of its separation pairs has both vertices incident to $f(G_C)$.
- *Invariant D:* No internal vertex of a graph $G_i \in \mathcal{G}$ belongs to a graph $G_j \in \mathcal{G}$, with $i \neq j$.
- *Invariant E:* For each graph $G_i \in \mathcal{G}$, all the vertices incident to $f(G_i)$ are assigned to G_i, except for two vertices, which are forbidden.
- *Invariant F:* Each vertex v incident to the outer face of a graph in $\mathcal{G}$ is assigned to at most one graph $G_i \in \mathcal{G}$. If a vertex v incident to the outer face of a graph in $\mathcal{G}$ is assigned to a graph $G_i \in \mathcal{G}$, then v is forbidden for all graphs $G_j \in \mathcal{G}$ such that v is incident to $f(G_j)$, with $j \neq i$.
- *Invariant G:* Each vertex assigned to a graph in $\mathcal{G}$ belongs to exactly one block of S.

During each step, we perform two different actions. Action 1 removes from $\mathcal{G}$ the only graph G_C which contains separation pairs, if such a graph exists, and partitions G_C into a set of triconnected planar graphs G_C^i to be added to $\mathcal{G}$. Action 2 removes from a graph $G_i \in \mathcal{G}$ a vertex incident to $f(G_i)$ and its incident edges and creates a new block to be added to S. At the end of each of the two actions, Invariants A–G are satisfied. The algorithm stops when $\mathcal{G}$ is empty, that is, when all the vertices of G are spanned by S.

A first difference between the triangulated and the non-triangulated case concerns the first step of the algorithm. Namely, while in the triangulated case we select one vertex v of the outer face and we initialize the cactus with the block composed of v and of its neighbors, in this new algorithm we initialize the cactus with the cycle delimiting the outer face.

Another important difference lies in Action 1, that is, in the way the graph G_C which may be not triconnected is partitioned into subgraphs. In the triangulated case, such a partition is done by considering the chords of $f(G_C)$. In the non-triangulated case we have to more generally consider separation pairs incident to $f(G_C)$, since we are not guaranteed that every two vertices composing a separation pair are joined by an edge. Refer to Fig. 23. The partition is performed by considering one separation pair at a time. At the beginning of every step of such an algorithm, we have a partition of G_C into a set of graphs G_i. Each graph G_i which still has a separation pair is further partitioned into two subgraphs G_i^1 and G_i^2 and each of the vertices incident to $f(G_i)$ is assigned to, or forbidden for, G_i^1 and G_i^2 by means of the same algorithm described in Sect. 4. Hence, the assignment of the vertices to the graphs G_i^1 and G_i^2 can be done in such a way that the invariant that each of G_i^1 and G_i^2 has at most two forbidden vertices is maintained. A dummy edge connecting the two vertices of the separation pair has to be added incident to the outer face of each of G_i^1 and G_i^2, if it does not exist yet, in order to maintain the invariant that all the vertices of the outer faces of G_i^1 and G_i^2 have already been assigned to some block of S. Such a dummy edge is incident to the outer faces of G_i^1 and G_i^2 and hence it will not be part of any new block that is added to S in the following steps of the algorithm. It is easy to see that the described procedure for partitioning a graph into subgraphs does not introduce new separation pairs, does not introduce multiple edges, and hence it terminates providing a set of triconnected planar graphs.

Concerning Action 2, while in the triangulated case at every step we add to the cactus either an edge or a triangulated cycle, in the non-triangulated case we add either an edge or a simple cycle. Such a cycle is obtained as follows (see Fig. 24). As in the triangulated case, consider a vertex v incident to the outer face of a subgraph $G_i \in \mathcal{G}$ and such that v is assigned to G_i. Consider the internal faces of G_i that are incident to v, except for the two faces f_1 and f_2 sharing an edge with $f(G_i)$. Add to S the cycle that passes through all the vertices that are incident to such faces. Remove vertex v and its incident edges from G_i, obtaining a new graph G_i^*. Consider the two vertices v_1' and v_1'' adjacent to v and belonging to f_1. A dummy edge (v_1', v_1'') is added to G_1^*, if it does not exist yet, incident to $f(G_i^*)$. Analogously, consider the two vertices v_2' and v_2'' adjacent to v and belonging to f_2 and add a dummy edge (v_2', v_2'') to G_i^*, if it does not exist yet, incident to $f(G_i^*)$. Such dummy edges allow to maintain the invariant that all the vertices incident to $f(G_i^*)$ have already been assigned to some block of S.

We choose to present the algorithm for triangulations as the main contribution of this paper because a proof of Conjecture 1 was very recently and independently presented by Leighton and Moitra at the *Symposium on Foundations*

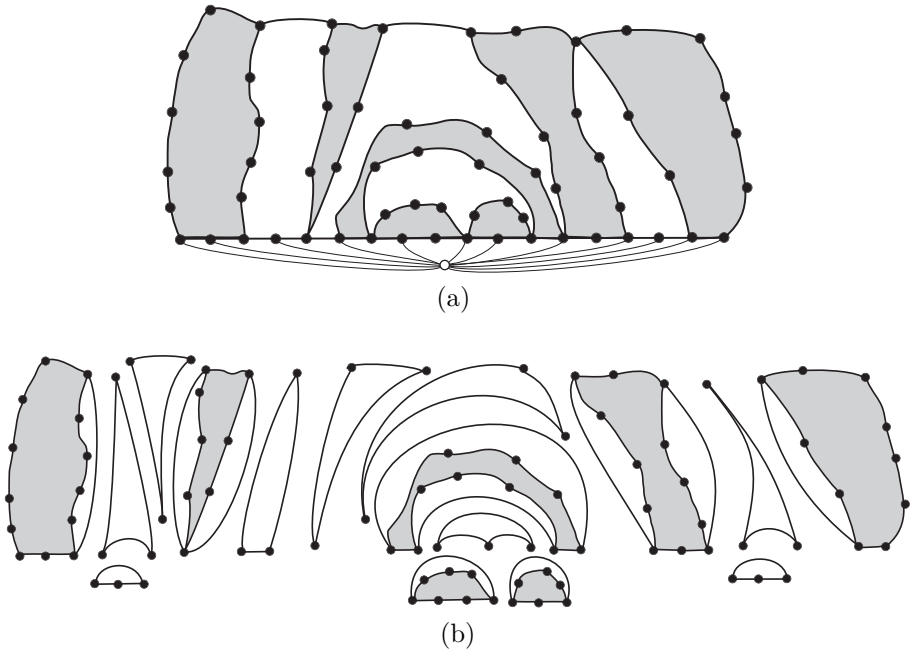


Figure 23: Partition of a biconnected graph having all separation pairs incident to the outer face into a set of triconnected planar graphs.

of Computer Science '08 [13]. Surprisingly, the approach used by Leighton and Moitra is exactly the same as ours. In fact, in [13] the authors define a class of graphs, called *Christmas cactus graphs*, which coincides with the class of non-triangulated binary cactuses; they show an algorithm to construct greedy drawings of Christmas cactus graphs and they show that every triconnected planar graph is spanned by a Christmas cactus graph. However, the way such results are achieved differs from ours. Such an issue is discussed below.

Concerning the geometric construction of greedy drawings of Christmas cactus graphs, the algorithm by Leighton and Moitra is quite similar to ours, even if a slightly different construction is used. Their algorithm places the nodes of the graph on a set of concentric circles $C_0, C_1, \ldots, C_k$, so that the block corresponding to the root ν of the BC-tree $\mathcal{T}$ has its nodes placed on C_0 and each block μ at depth i (where the depth is meant to be the number of B-nodes in the path from ν to μ in $\mathcal{T}$) is placed on C_i, except for the C-node parent of μ, which is placed on C_{i-1}. The difference between the radii of two consecutive circles (and hence the length of the edges of the drawing) exponentially decreases with i.

Concerning the construction of a Christmas cactus graph spanning a given triconnected planar graph, we have the main differences between our techniques and Leighton and Moitra's ones. In fact, in order to show that every triconnected planar graph is spanned by a Christmas cactus graph, they use some results from

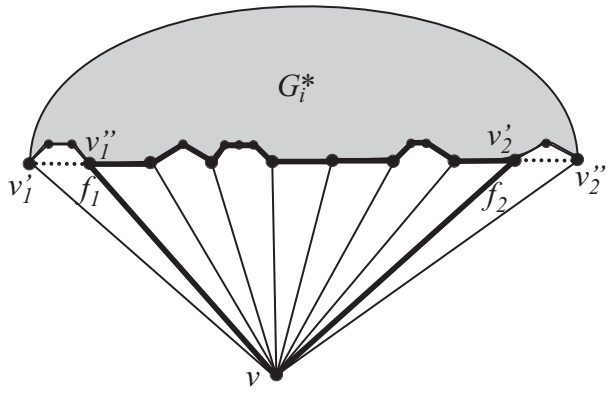


Figure 24: Action 2. The thick cycle is added to S. The dotted edges are inserted incident to $f(G_i^*)$, in order to maintain the invariant that all the vertices incident to $f(G_i^*)$ have already been assigned to some block of S.

a paper [8] by Gao and Richter.

Define a *circuit graph* to be an ordered pair (G, C) such that: (1) G is 2-connected and C is a polygon in G; (2) there exists an embedding of G in the plane such that C bounds a face; and (3) every separation pair of G has both vertices belonging to C. Hence, circuit graphs are a superclass of triconnected planar graphs. Define a *chain of blocks* $B_{i,1}, b_{i,1}, B_{i,2}, b_{i,2}, \ldots, B_{i,k_i-1}, b_{i,k_i-1}, B_{i,k_i}$ to be a connected graph such that each block contains at most two cutvertices and each cutvertex is shared by exactly two blocks.

In [8], Gao and Richter prove some strong structural results about circuit graphs, which are briefly described below. Gao and Richter prove that, given a circuit graph (G, C) and given two vertices x and y belonging to C, there exists a partition of $V(G)-V(C)$ into subsets $V_1, V_2, \ldots, V_m$ and there exist distinct vertices $v_1, v_2, \ldots, v_m \in V(C)-\{x, y\}$ such that, for $1 \leq i \leq m$: (i) the subgraph induced by $V_i \cup \{v_i\}$ is a chain of blocks $B_{i,1}, b_{i,1}, B_{i,2}, b_{i,2}, \ldots, B_{i,k_i-1}, b_{i,k_i-1}, B_{i,k_i}$, and (ii) $v_i \in V(B_{i,1}) \setminus \{b_{i,1}\}$.

Gao and Richter used this structural result in order to inductively prove that every triconnected planar graph (in fact, every circuit graph) has a closed 2-walk, which is a walk on the graph starting and ending at the same vertex and passing through each vertex of the graph at least once and at most twice.

The same result is used by Leighton and Moitra to inductively prove that, for every circuit graph (G, C) (and hence every triconnected planar graph G), a Christmas cactus graph S spanning G exists. In fact, the outline of their algorithm for spanning G consists of the following steps: (i) Use Gao and Richter's structural result in order to find chains of blocks $B_{i,1}, b_{i,1}, B_{i,2}, b_{i,2}, \ldots, B_{i,k_i-1}$, b_{i,k_i-1}, B_{i,k_i} spanning all the vertices of G not in C; (ii) inductively compute Christmas cactus graphs spanning each block $B_{i,j}$ (which is in turn a circuit graph); (iii) glue the Christmas cactus graphs spanning the blocks and C into a unique Christmas cactus graph spanning G.

Our spanning algorithm, as discussed above, finds the spanning graph of G without using Gao and Richter's result. Moreover, once one has a non-triangulated binary cactus spanning a triconnected planar graph G, it is easy to find a closed 2-walk that passes only through the edges of such a spanning graph. Hence, our algorithm for spanning triconnected planar graphs also provides a proof that every triconnected planar graph has a closed 2-walk alternative to Gao and Richter's one.

It is interesting to observe that our algorithm for spanning a triconnected planar graph with a non-triangulated binary cactus works more generally for circuit graphs (as the Leighton and Moitra's algorithm). In fact, in our algorithm, the only graph which may contain separation pairs before Action 1 is actually a circuit graph, since all its separation pairs are incident to the outer face. A spanning cactus for such a graph can hence be found with the same algorithm described above.

The main drawback of our algorithm (and of Leighton and Moitra's algorithm, as well) is that it uses real coordinates, hence it constructs drawings requiring exponential area once a finite resolution rule has been fixed. This leads to the following question:

Problem 1 *What are the area requirements of greedy drawings of triangulations and triconnected planar graphs?*

Several results related to the above problem have been recently shown. Namely, Eppstein and Goodrich [7] proved that every graph has a greedy drawing in the hyperbolic plane in which the vertex coordinates can be represented by $O(\log n)$ bits; Goodrich and Strash [10] proved that every triconnected planar graph has a greedy drawing in the Euclidean plane in which the vertex coordinates can be represented by $O(\log n)$ bits; Angelini *et al.* [1] proved that there exist trees requiring exponential area in any greedy drawing (that is, there exist trees that, in any greedy drawing in the Euclidean plane, require a polynomial number of bits to represent the Cartesian coordinates of the vertices).

While every triconnected planar graph admits a greedy drawing, not all biconnected planar graphs and not all trees admit a greedy drawing. For example, in [13] it is shown that a complete binary tree with 31 nodes does not admit any greedy drawing. Hence, the following problem is worth studying:

Problem 2 *Characterize the class of trees (resp. of biconnected planar graphs) that admit a greedy drawing.*

In the conference version [14] of [15], Papadimitriou and Ratajczak conjecture that every triconnected planar graph admits a *convex greedy drawing*, that is, a greedy drawing that is also planar and convex. Although some partial positive results are known [6, 9], the following problem is still open:

Problem 3 *Does every triconnected planar graph admit a convex greedy drawing?*

Finally, most of the known algorithms for constructing greedy graph drawings rely on the knowledge of the entire graph topology. Designing distributed algorithms for computing greedy drawings or proving that such algorithms do not exist would be theoretically interesting and useful in practice for greedy routing.

Acknowledgements

Thanks to Tom Leighton and Ankur Moitra for providing us with their paper, for introducing us to Gao and Richter's results, and for helping us to clarify the relationship between our results, their own, and Gao and Richter's ones.

References

[1] P. Angelini, G. Di Battista, and F. Frati. Succinct greedy drawings do not always exist. In D. Eppstein and E. Gansner, editors, *International Symposium on Graph Drawing (GD '09)*, LNCS, 2009. to appear.

[2] P. Angelini, F. Frati, and L. Grilli. An algorithm to construct greedy drawings of triangulations. In I. G. Tollis and M. Patrignani, editors, *International Symposium on Graph Drawing (GD '08)*, volume 5417 of *LNCS*, pages 26–37, 2008.

[3] M. Ben-Chen, C. Gotsman, and S. J. Gortler. Routing with guaranteed delivery on virtual coordinates. In *Canadian Conference on Computational Geometry (CCCG '06)*, 2006.

[4] M. Ben-Chen, C. Gotsman, and C. Wormser. Distributed computation of virtual coordinates. In J. Erickson, editor, *ACM Symposium on Computational Geometry (SoCG '07)*, pages 210–219, 2007.

[5] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

[6] R. Dhandapani. Greedy drawings of triangulations. In S.-T. Huang, editor, *SIAM Symposium on Discrete Algorithms (SODA '08)*, pages 102–111, 2008.

[7] D. Eppstein and M. T. Goodrich. Succinct greedy graph drawing in the hyperbolic plane. In I. G. Tollis and M. Patrignani, editors, *International Symposium on Graph Drawing (GD '08)*, volume 5417 of *LNCS*, pages 14–25, 2008.

[8] Z. Gao and R. B. Richter. 2-walks in circuit graphs. *Journal of Combinatorial Theory, Series B*, 62(2):259–267, 1994.

[9] S. K. Ghosh and K. Sinha. On convex greedy embedding conjecture for 3-connected planar graphs. In M. Kutylowski, W. Charatonik, and M. Gebala, editors, *Symposium on Fundamentals of Computation Theory (FCT '09)*, volume 5699 of *LNCS*, pages 145–156, 2009.

[10] M. T. Goodrich and D. Strash. Succinct greedy geometric routing in the euclidean plane. In Y. Dong, D.-Z. Du, and O. Ibarra, editors, *International Symposium on Algorithms and Computation (ISAAC '09)*, LNCS, 2009. to appear.

[11] R. Kleinberg. Geographic routing using hyperbolic space. In *IEEE Conference on Computer Communications (INFOCOM '07)*, pages 1902–1909, 2007.

[12] B. Knaster, C. Kuratowski, and C. Mazurkiewicz. Ein beweis des fixpunktsatzes fur n dimensionale simplexe. *Fundamenta Mathematicae*, 14:132–137, 1929.

[13] T. Leighton and A. Moitra. Some results on greedy embeddings in metric spaces. In *IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 337–346, 2008.

[14] C. H. Papadimitriou and D. Ratajczak. On a conjecture related to geometric routing. In *Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS '04)*, volume 3121 of *LNCS*, pages 9–17, 2004.

[15] C. H. Papadimitriou and D. Ratajczak. On a conjecture related to geometric routing. *Theoretical Computer Science*, 344(1):3–14, 2005.

[16] A. Rao, C. H. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In D. B. Johnson, A. D. Joseph, and N. H. Vaidya, editors, *ACM International Conference on Mobile Computing and Networking (MOBICOM '03)*, pages 96–108, 2003.

[17] W. Schnyder. Embedding planar graphs on the grid. In *SIAM Symposium on Discrete Algorithms (SODA '90)*, pages 138–148, 1990.