Distributed Computation of Virtual Coordinates

Mirela Ben Chen, Craig Gotsman, Camille Wormser

Distributed Computation of Virtual Coordinates

Mirela Ben-Chen Craig Gotsman Camille Wormser

Abstract

Sensor networks are emerging as a paradigm for future computing, but pose a number of challenges in the fields of networking and distributed computation. One challenge is to devise a *greedy routing* protocol – one that routes messages through the network using only information available at a node or its neighbors. Modeling the connectivity graph of a sensor network as a 3-connected planar graph, we describe how to compute on the network in a distributed and local manner a special geometric embedding of the graph. This embedding supports a geometric routing protocol based on the "virtual" coordinates of the nodes derived from the embedding.

1 Introduction

Sensor networks are a collection of (usually miniature) devices, each with limited computing and wireless communication capabilities, distributed over a physical area. The sensor network collects data from its environment and should be able to integrate it and answer queries related to this data. Sensor networks are becoming more and more attractive in many application domains.

The advent of sensor networks has posed a number of research challenges to the networking and distributed computation communities. Since each sensor can typically communicate only with a small number of other sensors within a short range, information generated at one sensor can reach another sensor only by routing it through the network. Traditional routing algorithms rely only on the combinatorial connectivity graph of the network, but the introduction of so-called *location-aware* sensors, namely, those that also *know* what their physical location is (e.g. by using a GPS receiver), permit more efficient *geographic* or *geometric* routing.

In geometric routing we consider the following problem: A packet is to be routed across the network from a source sensor to a destination sensor. The physical locations – the *coordinates* – of the source and destination sensors are known. When a sensor receives a packet, it must decide to which of its *neighbors* it should forward the packet based on a *local* decision. By local decision, we mean that the decision is made based *only* on local information - the coordinates of the current sensor, the destination, and the sensor's neighbors. Despite this restrictive locality, the routing algorithm should guarantee that the packet will indeed arrive at the destination.

One simple geometric routing scheme is *greedy routing*. In greedy routing, when a sensor receives a packet, it forwards the packet to the neighbor that is *closest* in some sense to the destination sensor. The main problem with greedy routing is that it may encounter local minima, also known as *routing voids* or *holes*, when the current sensor has no neighbor closer to the destination than itself. When such a local minimum is encountered, the packet is "stuck", greedy routing cannot continue, and the delivery fails. Examples of greedy routing are greedy Euclidean routing, which is based on Euclidean distance to the destination, or compass routing, based on angular distance to the destination [10].

Greedy routing is based on the real (physical) coordinates of the network (as reported by the GPS system), but there is nothing special about these coordinates, in the sense that any other coordinate system may be used. Thus, it may be advantageous to endow the sensors with new "virtual" coordinates, which will behave better in the greedy routing scenario. Algorithms that generate and use virtual coordinates can be found in the literature [11, 12, 13]. Unfortunately, very few of them can guarantee that routing using the virtual coordinates will never fail in general settings.

Greedy routing on a connectivity graph of a sensor network raises a number of interesting theoretical questions: What are the virtual coordinates that will support greedy routing? Is such a coordinate system even guaranteed to exist? What is the smallest dimension of space in which such a coordinate system exists? We call such good embeddings *greedy embeddings*. The dimension of the embedding space dictates the amount of space that must be allocated per sensor for storing the node coordinates. Ideally, this should be a small constant, typically 2 or 3. Some theoretical questions related to this type of low-dimensional embedding were partially answered by Papadimitriou and Ratajczak [12]. They conjectured that any 3-connected planar graph has a greedy Euclidean embedding (i.e. an embedding which supports greedy Euclidean routing) in $\mathbb{R}^2$. Should the conjecture be true, this is an important result for sensor networks, since most communication graphs will have a 3-connected planar subgraph. A possible way to use these coordinates is to find the relevant subgraph, compute the greedy coordinates, and then discard the planar subgraph, retaining the virtual coordinates. Greedy Euclidean routing on the full communication graph using the same virtual coordinates is still guaranteed to succeed. Papadimitriou and Ratajczak provided a number of geometric characterizations of greedy Euclidean embeddings in $\mathbb{R}^2$. A notable sufficient (but not necessary) condition is that all angles of the straight-line plane drawing are less than 120°. It is also quite easy to show [12] that a Delaunay triangulation is a greedy Euclidean embedding.

Recent work by Ben-Chen, Gotsman and Gortler [1] also considered greedy routing on 3-connected planar graphs. Their most interesting result may be thought of as midway between the easy observation that a Delaunay triangulation is a greedy Euclidean embedding of a triangulated graph (if such an embedding is possible [5, 7]), and the conjecture of Papadimitriou and Ratajczak [12] that every 3-connected planar graph has a greedy Euclidean embedding. Recall that the Delaunay triangulation is the orthogonal dual of the Voronoi diagram. Instead of using Voronoi diagrams, Ben-Chen et al. proposed to use *power diagrams*. These are generalizations of the Voronoi diagram, where each site is endowed with a radius, and the distance to a site is measured using the *power distance* – which takes the radii into account (a Voronoi diagram is a special case of the power diagram where all radii are equal). Power diagrams where each cell contains its site are called *contained power diagrams*, and Ben-Chen et al. showed that the duals of these embeddings support *greedy power routing* – greedy routing using the (non-Euclidean) power distance. The resulting challenge is to compute planar coordinates and a radius for each of the vertices of a given 3-connected planar graph, such that resulting power diagram is contained and its combinatorial dual is isomorphic to the given graph. A special case of such an embedding is that obtained from the so-called *coin-graph* embedding [15, 16] (where the radii are those of the coins). It is easy to see that the edges of the dual are tangent to the coins at their intersection points.

The focus of this paper is the *computation* of a greedy power embedding in a *local* manner. Just as the actual routing of a message should be done locally, so the embedding on which the routing is based should be computed in a distributed manner by the sensors in the network, each communicating *only* with its neighbors in the network connectivity graph. Our starting point is the algorithm proposed by Thurston in 1985 [16, 17] for computing coin-graph embeddings (see Collins and Stephenson [4] for a practical implementation): it is an iterative process that computes a set of radii that converge to the desired values. The radius associated with a node is modified at each step based on a certain sum of angles around the node, a value depending only on the node and its immediate neighbors, thus locally computable. The algorithm terminates when this sum reaches 2π (up to some numerical tolerance) at all nodes. Once the radii have been computed, the embedding may be computed easily by an incremental layout process. Thus these special embeddings may be computed locally by a sensor network. They are however, quite restrictive, and we address here the question of how to compute an embedding corresponding to a member of the broader class of contained power diagrams.

Towards this end, we adopt the Thurston embedding algorithm, but replace its termination conditions by geometric and local ones. We demonstrate that these new termination conditions allow us to stop the iterations as soon as we can guarantee that the routing will deliver, minimizing the amount of computations. By minimizing the number of computations and distributing the computation among the vertices of the graph, this algorithm is especially suitable for a distributed implementation over a sensor network with limited computation resources, allowing it to compute its virtual coordinates by itself.

2 Previous Work

As mentioned in the Introduction, any Delaunay triangulation is a greedy Euclidean embedding. Thus the following natural question arises: Given a triangulated planar subgraph of the communication graph of a sensor network, can we embed the subgraph in the plane such that the resulting triangulation will be Delaunay? Such a process is called *Delaunay realization*. Moreover, can it be done in a local manner by computing at the nodes of the graph? Dillencourt and Smith [5] showed that not all triangulated planar graphs are Delaunay-realizable, and the class of Delaunay-realizable graphs is essentially equivalent to the class of inscribable graphs – ones that may be embedded as a convex polyhedron in $\mathbb{R}^3$ with vertices on the sphere. Complete characterizations of the Delaunay realizability of a planar triangulated graph have been given by Hodgson et al.[8] and Hiroshima et al.[7]. This involves defining a so-called *coherent angle system* for the edges. Experiments run by Hiroshima et al.[7] showed that the vast majority of the set of planar triangle graphs are Delaunay realizable. Despite this, an algorithm to actually compute the embedding is quite difficult. It is related to another difficult embedding problem, that of generating a coin-graph embedding. The latter was solved using an iterative algorithm by Thurston [16], and solved in a more general setting, where the circles have prescribed *intersection* angles, as a global optimization problem by Bobenko and Springborn [2]. Both algorithms solve for the radii of the circles. The same algorithm of Bobenko and Springborn may be used to perform Delaunay realization by solving for the radii of the circumcircles of the Delaunay triangles, using a previously computed coherent angle system [9].

Papadimitriou and Ratajczak [12] studied the problem of generating greedy Euclidean embeddings for 3-connected planar graphs. While unable to prove the existence of such embeddings, they showed that the following two conditions are equivalent.

1. An embedding $p : V \rightarrow \mathbb{R}^2$ is a greedy Euclidean embedding.
2. Denote by $\mathrm{Cell}_G(v)$ the cell associated with site $p(v)$ in the "local" Voronoi diagram of just the sites $\{p(v)\} \cup \{p(w) : w \in N_G(v)\}$, where N_G denotes the neighbors of v in G. Then $\forall v \in V,\ p(w) \in \mathrm{Cell}_G(v)$ iff $w = v$.

Note that one direction of the equivalence in condition 2, namely $p(v) \in \mathrm{Cell}_G(v)$ is trivial. The challenge is that $\mathrm{Cell}_G(v)$ contains *only* $p(v)$, and is void of other sites. Note also that although $\mathrm{Cell}_G(v)$ may be constructed locally (based only on the positions of v's neighbors), checking this condition cannot be done locally by v, since v must check that *all other nodes* are not in its cell.

Ben-Chen et al.[1] described how to perform greedy power routing using duals of *contained* power diagrams. A power diagram associates with each site $p(v)$ a radius $r(v)$, and the distance of a point $q \in \mathbb{R}^2$ from $p(v)$ is defined as $d(q,p(v))^2 = e(q,p(v))^2 - r(v)^2$, where $e(\cdot,\cdot)$ is the Euclidean metric. The power cell $\mathrm{Cell}(v)$ is the (convex) region of points q such that $d(q,p(v)) \leq d(q,p(w))$ for all $w \neq v$. A contained power diagram is one where $p(v) \in \mathrm{Cell}(v)$ for all v. Note that, in contrast to the cell defined by Papadimitriou and Ratajcazk, this cell may *not* be constructed locally, since it may depend on sites not neighboring v in the connectivity graph. However, once it is constructed, the containedness property may be *easily* checked locally. Thus a key objective of this paper is to formulate a *local* condition for checking that the adjacency graph of the power diagram is indeed G (or not much different from it, in the non-triangulation case) and checking containedness. This, when applied as a local termination condition to the Thurston algorithm, allows to generate greedy power embeddings in a distributed manner on a sensor network.

3 Triangulated Graphs

Let $G(V,E)$ be a combinatorial triangulation. We assume that G is planar and we denote by B its boundary, which is a cycle. In the following, we study a map $\phi : V \rightarrow \mathbb{D}^2 \times \mathbb{R}$, which associates to each vertex v a point $p(v)$ in the unit disk and a scalar weight $\sigma(v)$. We denote by $\mathrm{Conv}(p(V))$ the convex hull of the associated points.

Definition 3.1 *A power diagram is said to be* contained *if each site is inside its cell (see Figure* **??***).*

Let us now recall a slightly stronger version of the result of Ben Chen et al.[1] that we will use to provide a sufficient condition for the greedy power routing to deliver.

Theorem 3.2 (Ben Chen et al.) *If the restriction of the power diagram of $\phi(V)$ to $\mathrm{Conv}(p(V))$ is contained and if its adjacency graph (i.e. the combinatorial dual) is a subgraph of G, then greedy power routing delivers on ϕ.*

Proof See the proof by Ben-Chen et al.[1]. Just take into account that the only edges of the power diagram that matter in the proof are the ones that are inside the convex hull of the points, and that the proof does not change if only a subgraph of G is obtained as the adjacency graph of the power diagram. □

Definition 3.3 *If $w_1, \ldots, w_n$ are the neighbors of v in G, the* local cell *of v in G, denoted by $\mathrm{Cell}_G(v)$, is the cell of v in the power diagram of $\{\phi(v), \phi(w_1), \ldots, \phi(w_n)\}$ (see Figure* **??***).*

In the following definition, when we refer to the order of vertices around another vertex, we mean the cyclic order of vertices, which is independent of the embedding in the case of a triangulation (except that we can reverse all orientations).

Definition 3.4 *For any vertex $v \in V$, we say that property* $\mathrm{LPD}(v, \phi)$ *(Local Power Diagram) is satisfied if and only if*

- *if $w_1, \ldots, w_n$ are the neighbors of v in G (in this order), then the cell $\mathrm{Cell}_G(v)$ contains $p(v)$ and the cells adjacent to it are exactly the ones of $w_1, \ldots, w_n$ (in this order, see Figure* **??***);*
- *Let $v \in B$. Denote by w_1 and w_n the two neighbors of v that belong to B and that are linked to v by boundary edges. Then in the power diagram of $\{\phi(v), \phi(w_1), \ldots, \phi(w_n)\}$, $\mathrm{Cell}(v) \cap \mathrm{Cell}(w_1) \cap \mathrm{Cell}(w_n)$ is either empty (which means that $\mathrm{Cell}(v)$ is unbounded) or it is a point outside the unit disk $\mathbb{D}^2$.*

Note that the condition about the order of neighbor cells around a given cell is equivalent to requiring that the graph is properly embedded (this follows from the convexity of the power diagram cells). Thus, if G is known to be embedded, specifying the order of neighbor cells is not necessary.

Theorem 3.5 *If*

$$\forall v \in V, \quad \mathrm{LPD}(v, \phi),$$

then the restriction of the power diagram of $\phi(V)$ to $\mathrm{Conv}(p(V))$ is contained and its adjacency graph is G.

Proof From now on, we denote by $\mathrm{Cell}(w)$ the cell of $\phi(w)$ in the power diagram of $\phi(V)$, and by $\mathrm{Cell}^*_v(w)$ the cell of w in the power diagram of $\{\phi(v), \phi(w_1), \ldots, \phi(w_n)\}$, where $w_1, \ldots, w_n$ are the neighbors of v in G. Let ρ be the restriction to $\mathrm{Conv}(p(V))$.

Let us now prove that we have in fact $\rho(\mathrm{Cell}^*_v(v)) = \rho(\mathrm{Cell}(v))$ for all $v \in V$. First note that since $\mathrm{Cell}(v) \subset \mathrm{Cell}^*_v(v)$ for all $v \in V$, $\cup_{v \in V} \rho(\mathrm{Cell}^*_v(v)) = \mathrm{Conv}(p(V))$.

For each vertex $v \in V$, we consider the usual lifting $\ell_v : x \mapsto 2(x|v) - \phi(v) + \sigma(v)^2$. The power diagram of $\phi(V)$ is the projection of the upper envelope of the hyperplanes $\ell_v(\mathbb{R}^2)$. We now show that the $\ell_v(\rho(\mathrm{Cell}(v)))$ can be glued into a convex terrain over the convex domain $\mathrm{Conv}(p(V))$.

If v and w are neighbors in G and $v \notin B$, let p and q be the two vertices incident to the edge (v, w). Let α be the power diagram vertex defined by v, w and p, and let β be the power diagram vertex defined by v, w and q. The hypotheses $\mathrm{LPD}(v, \phi)$ and $\mathrm{LPD}(w, \phi)$ imply that the segment $[\alpha\beta]$ is an edge common to $\mathrm{Cell}^*_v(v)$ and $\mathrm{Cell}^*_w(w)$ because the four vertices v, w, p and q will all appear in the computations of the border of both cells.

This implies that $\ell_v(\mathrm{Cell}^*_v(v))$ and $\ell_w(\mathrm{Cell}^*_w(w))$ can be glued together along their common edge $\ell_v([\alpha\beta]) = \ell_w([\alpha\beta]) = [AB]$. Furthermore, by looking at the local diagram of v and its neighbors, one can see that the angle between $\ell_v(\mathrm{Cell}^*_v(v))$ and $\ell_v(\mathrm{Cell}^*_w(w))$ along $[AB]$ is convex.

Let us now consider the case where both v and w are boundary vertices. Let p be the incident vertex to (v, w) in G and consider the edge $e(v, w) = \mathrm{Cell}^*_v(v) \cap \mathrm{Cell}^*_v(w)$. Hypothesis $\mathrm{LPD}(v, \phi)$ implies that this edge $e(v, w)$, whether infinite or not, has only one vertex inside the unit disk $\mathbb{D}^2$, which is the power diagram vertex defined by v, w and p. We also know that $e(v, w)$ is orthogonal to the line $(p(v)p(w))$ and that e reaches the boundary of $\mathbb{D}^2$. By symmetry, $e(w, v)$ has the same properties. It follows that $\rho(e(v, w)) = \rho(e(w, v))$. This proves again that $\ell_v(\mathrm{Cell}^*_v(v))$ and $\ell_w(\mathrm{Cell}^*_w(w))$ can be glued together along a convex edge.

Finally, we obtain that the $\ell_v(\mathrm{Cell}^*_v(v))$ can be glued together into a locally convex polyhedral terrain $\mathcal{P}$ over the convex domain $\mathrm{Conv}(p(V))$. It follows that $\mathcal{P}$ is globally convex and is in fact the restriction of a convex polytope and that the projection of its edges onto $\mathrm{Conv}(p(V))$ is a restricted power diagram, whose sites happen to be the elements of $\phi(V)$, by construction. The way the patches have been glued together shows that the adjacency graph of this restricted power diagram is exactly G. $\square$

We can now state the following corollary of Theorems 3.2 and 3.5:

Corollary 3.6 *If*

$$\forall v \in V, \quad \mathrm{LPD}(v, \phi),$$

then greedy power routing delivers on ϕ.

4 General Distances

Papadimitriou and Ratajczak [12] have provided a geometric condition on embeddings of 3-connected planar graphs which characterizes greedy Euclidean embeddings. We now present this result in the more general context of arbitrary distance functions, and explain how it relates to Section 3. We will need this for the extension of the results of Section 3 to more general planar graphs.

Given a field d of distance functions $\{d_x : \mathbb{R}^2 \to \mathbb{R}, x \in \mathbb{R}^2\}$ (these functions are arbitrary real functions) and a set of sites $V \subset \mathbb{R}^2$, we can define two kinds of distance diagrams:

- the usual one, where the cell of a site v is defined as

$$\mathrm{Cell}(v) = \{x \in \mathbb{R}^2, d_v(x) \leq d_w(x), \forall w \in V\}$$

- the reciprocal one, where the cell of a site v, called the *reciprocal cell* is defined as

$$\mathrm{Cell}^\circ(v) = \{x \in \mathbb{R}^2, d_x(v) \leq d_x(w), \forall w \in V\}$$

Note that in the first case, the computation of a cell depends only on the distance functions of the sites. In contrast, in the second case, it depends on the distance functions at each point in the plane. Thus, the reciprocal diagram is usually impossible to compute (locally) if the distance functions are too general.

Just as we defined the local cell $\mathrm{Cell}_G(v)$, we can define the local *reciprocal* cell $\mathrm{Cell}^\circ_G(v)$ and state a generalized version of the result of Papadimitriou and Ratajczak:

Theorem 4.1 *Given a field d of distance functions $\{d_x : \mathbb{R}^2 \to \mathbb{R}, x \in \mathbb{R}^2\}$, greedy routing on a graph $G(V, E)$ with respect to d delivers if and only if for each vertex $v \in V$, the local reciprocal cell $\mathrm{Cell}^\circ_G(v)$ contains no vertex other than v.*

Proof The proof is exactly the same as the one given by Papadimitriou and Ratajczak [12]. $\square$

This is not a practical result. However, in the case of symmetrical distance functions, i.e. distance functions such that $\forall x, y \in \mathbb{R}^2, d_x(y) = d_y(x)$, the usual cell and the reciprocal cell are identical, namely $\mathrm{Cell} = \mathrm{Cell}^\circ$ and $\mathrm{Cell}_G = \mathrm{Cell}^\circ_G$. This is the case not only for the Euclidean distance, but also for the power distance: each point x in the plane is endowed with an arbitrary radius r_x, and the distance between two points x and y is defined as $d_x(y) = d_y(x) = \|x - y\|^2 - r_x^2 - r_y^2$ (if x is not a site, we may choose $r_x = 0$ or any arbitrary real value). Thus, we can now make use of the following version of the theorem:

Theorem 4.2 *Greedy power routing delivers if and only if for each vertex $v \in V$, the local cell $\mathrm{Cell}_G(v)$ for the power distance contains no vertex other than $p(v)$ (see Figure* **??***).*

We summarize our results so far in the following diagram, which details the links between the various conditions. These hold for both Euclidean and power distances:

$$\begin{array}{ccc}
\forall v \in V, \quad \mathrm{LPD}(v, \phi) & \overset{\text{Theorem 3.5}}{\Longleftrightarrow} & \forall v \in V, \quad v \in \mathrm{Cell}_G(v) = \mathrm{Cell}(v) \\
 & & \text{Theorem 3.2} \quad \Downarrow \\
\text{Greedy routing delivers on } \phi(G) & \underset{\text{Theorem 4.2}}{\Longleftrightarrow} & \forall v \in V, \quad \mathrm{Cell}_G(v) \text{ contains only } v
\end{array}$$

Note that the upper right condition may also be stated as "G is the dual graph of the contained distance (power or Voronoi) diagram of $\phi(V)$."

5 3-Connected Planar Graphs

Let us now consider the more general case of a 3-connected planar graph. In the previous section, we proved that satisfying LPD at every vertex implied that the power diagram of $\phi(V)$ admitted G as its adjacency graph. This cannot be the case if G is not a triangulation: such graph can only be the dual graph of a degenerate power diagram, which would be unstable under perturbation of the vertices, whereas LPD is stable.

In order to state the next definition, we need to recall the following result:

Lemma 5.1 *If a set of points $\{p_1, \ldots, p_n\}$ is in convex position, for any radii $(r_i)_{1\leq i\leq n}$, the adjacency graph of the power diagram of the circles $\mathcal{C}(p_i, r_i)$ is a triangulation of* $\mathrm{Conv}(\{p_1, \ldots, p_n\})$.

Proof The dual of a power diagram is known to be an embedded triangulation, called the regular triangulation. However, in order to have a triangulation of $\mathrm{Conv}(\{p_1, \ldots, p_n\})$, each point p_i has to appear as a vertex of this triangulation. In other words, it has to have a non-empty cell, which is guaranteed by the convexity assumption. □

Definition 5.2 *If p is a convex embedding of G, the ϕ-*triangulation of *G is defined in the following way: if f is a non-triangle face, $p(f)$ is convex and we glue along f the dual graph of the power diagram of the vertices of f, which is indeed a triangulation of f, thanks to Lemma 5.1. The resulting triangulation of G is called the ϕ-triangulation of G and is denoted by $G(\phi)$ (see Figure* **??***).*

In case we are in a degenerate configuration, we choose a triangulation obtained after some infinitesimal perturbation.

We are now able to present the generalized version of the condition that we proved sufficient in the triangulated case:

Definition 5.3 *For any vertex $v \in V$, we say that property* $\mathrm{GLPD}(v, \phi)$ *(Generalized Local Power Diagram) is satisfied if and only if the faces incident to v are convex,* $\mathrm{LPD}(v, \phi)$ *is satisfied in $G(\phi)$ and for each non-triangle face $f = (v, w_1, \ldots, w_n)$ incident to v, the local cell* $\mathrm{Cell}_G(v)$ *of v in G intersects f only along segments $[w_n v]$ and $[v w_1]$ (see Figure* **??***).*

Note that, in the last condition, the local cell is computed in G, and not in $G(\phi)$: otherwise, the condition is trivially satisfied.

Theorem 5.4 *If p is a convex embedding and*

$$\forall v \in V, \quad \mathrm{GLPD}(v, \phi),$$

then each local cell $\mathrm{Cell}_G(v)$ *contains only its site $p(v)$.*

Proof From the proof of Theorem 3.5, we know that LPD being satisfied for $G(\phi)$ at every vertex implies that the local cell $\mathrm{Cell}_{G(\phi)}(v)$ computed in $G(\phi)$ is exactly the cell of the power diagram of $\phi(V)$, and that this diagram is a contained embedding of $G(\phi)$.

We need the local cell $\mathrm{Cell}_G(v)$ computed in G to be empty of other vertices. The cell $\mathrm{Cell}_G(v)$ contains $\mathrm{Cell}_{G(\phi)}(v)$. We prove now that the difference $\mathrm{Cell}_G(v) \setminus \mathrm{Cell}_{G(\phi)}(v)$ is included in the union of the faces incident to v. Note that $\mathrm{Cell}_{G(\phi)}(v)$ is not itself included in this union.

Let us consider now a non-triangle face $f = (v, w_1, \ldots, w_n)$ incident to v. We denote by $W_f = \{w_{i_1}, \ldots, w_{i_k}\}$ the set of vertices of f that belong to $W = N_{G(\phi)}(v) \setminus N_G(v)$. The cell of v in the power diagram of $\{v\} \cup N_G(v) \cup W_f$ is denoted by $\mathrm{Cell}_f(v)$.

By convexity of f, and using the fact the the local cells of the w_i are not allowed to cross f along the segments $[w_n v]$ and $[v w_1]$, one can easily see that $\mathrm{Cell}_G(v) \setminus \mathrm{Cell}_f(v)$ is included in f. Since $\mathrm{Cell}_{G(\phi)}(v) = \cap_f \mathrm{Cell}_f(v)$, where the intersection is taken over all non-triangle faces f incident to v, the result follows. □

One could wonder why we do not impose the stronger condition that triangle faces should satisfy the same property as non-triangle faces. The reason is that this condition is not equivalent to LPD in the triangulated case, whereas GLPD is. Since we want a condition as weak as possible, we avoid this.

The following corollary is a consequence of Theorems 4.2 and 5.4:

Corollary 5.5 *Under the conditions that p is a convex embedding and*

$$\forall v \in V, \quad \mathrm{GLPD}(v, \phi),$$

greedy power routing delivers on ϕ.

6 Circle Packings

Ultimately, we would like to use the LPD and GLPD conditions as a local termination condition for generating embeddings whose duals are contained power diagrams, using the Thurston algorithm, which was originally designed for generating coin-graph embeddings. Towards this end, we first prove that coin-graph embeddings of G satisfy LPD or GLPD.

Definition 6.1 *Given a planar triangulation $G(V, E)$, a G-*circle packing *is a set $\mathcal{C}$ of circles in the plane with a bijection $\gamma : V \to \mathcal{C}$ such that $\gamma(v)$ and $\gamma(w)$ are externally tangent if and only if $\{v, w\}$ is an edge of G.*

Definition 6.2 *A G-circle packing is said to be* locally univalent *if for any vertex $v \in V$, the circles corresponding to v and to its neighbors in G have mutually disjoint interiors.*

Let us present a few important results about these circle packings. A detailed presentation of the subject can be found in Stephenson [15].

Theorem 6.3 ([15], p. 18) *Given any assignment of positive radii to the boundary vertices of G, there exists (in the euclidean and in the hyperbolic plane) an essentially unique locally univalent circle packing for G whose boundary circles have these numbers as their radii.*

Essentially unique is to be understood as up to isometry.

Definition 6.4 *A G-circle packing is said to be* univalent *if its circles have mutually disjoint interiors.*

In the sequel, we will need circle packings that are univalent. Thus, we will use the following result:

Theorem 6.5 ([15], page 62) *Let G be a combinatorial closed disc (that is, simply connected, finite, and with nonempty boundary). Then there exists an essentially unique univalent circle packing $\mathcal{P}_G$ included in the unit disc such that any boundary circle is internally tangent to the unit disc.*

We will refer to this kind of packing as a *G-circle packing of the unit disc.*

Note that the previous results are stated for a triangulated graph. However, these two theorems are still true for 3-connected planar graphs, if a rigidity condition is added to the definition of circle packing:

Definition 6.6 *Given a 3-connected planar graph $G(V, E)$, a G-*circle packing *is a set $\mathcal{C}$ of circles in the plane with a bijection $\gamma : V \to \mathcal{C}$ such that $\gamma(v)$ and $\gamma(w)$ are externally tangent if and only if $\{v, w\}$ is an edge of G, and such that for each face $f = (w_1, \ldots, w_n)$ of G, there exists a circle $c(f)$ which is orthogonal to all circles $\gamma(w_i)$, $1 \le i \le n$.*

This last definition allows to state the following result for general 3-connected planar graphs.

Theorem 6.7 *If G is a planar triangulation and if $\phi(G)$ is a G-circle packing of the unit disc, then*

$$\forall v \in V, \quad \mathrm{LPD}(v, \phi)$$

Proof Since the bisector between two tangent circles is their common tangent line, we obtain that the local cell of a circle is the intersection of the halfspaces delimited by some tangent lines. The result follows. □

Theorem 6.8 *If G is a 3-connected planar graph and if $\phi(G)$ is a G-circle packing of the unit disc, then we have*

$$\forall v \in V, \quad \mathrm{GLPD}(v, \phi)$$

Proof Let f be a face of G. By definition of the G-circle packing, there exists a circle $c(f)$ which is orthogonal to the circles of the vertices of f. It follows that c_f is inscribed in f, thus p is a convex embedding. We are in fact in the most degenerate case, and the faces can be triangulated arbitrarily to obtain a ϕ-triangulation of G. However, whichever triangulation we choose, the power diagram face of v is the polygon whose vertices are the centers of circles c_f, for the faces f incident to v. The result easily follows. □

7 The Algorithm

We now derive from Sections 3 and 5 a distributed algorithm for the computation of virtual coordinates that allow the greedy power routing algorithm to deliver. The correctness of the algorithm follows from Section 6, since, in the worst case, the conditions LPD (or GLPD) will be satisfied when the algorithm converges to a coin-graph embedding, which is guaranteed.

7.1 The Thurston Algorithm

We present in this section the algorithm that Thurston designed for the numerical computation of coin-graph embeddings (so called *circle packings*).

The Thurston algorithm consists of setting the value of the boundary radii and updating all internal radii in order to satisfy local univalence. This step is repeated until some error bound on the local univalence error (measured as an angular error) is reached. At this point, a layout process is required to translate the radii values into planar coordinates of the centers. The convergence of this process to a locally univalent circle packing, in the Euclidean and hyperbolic case, is proved in [3]. See Collins and Stephenson [4] for a practical and efficient implementation of this algorithm. In order to guarantee that LPD or GLPD is satisfied by the circle packing obtained by such process, we perform the computations in the hyperbolic plane, with infinite boundary radii. This will give us a globally univalent circle packing of the unit disc, thanks to Theorem 6.5. Theorems 6.7 and 6.8 then show that LPD or GLPD are satisfied.

Note that this algorithm works for triangulations only. However, it can be generalized to more general 3-connected planar graphs, with the additional constraint specified in Definition 6.6.

In the following, we represent the Thurston algorithm by a sequence of so-called *circle mapping functions* $(\phi_n)_{n\in\mathbb{N}}$ that map vertices of V to circles in the plane. The distance between two such functions is measured as the Euclidean distance d on $\mathbb{R}^{3|V|}$. We denote by Φ_G the function that maps the vertices to the limit circle packing Φ_G, which is unique up to some Möbius transformation.

7.2 Termination

Our algorithm consists of starting the Thurston algorithm to compute a circle packing in the Poincaré model of the hyperbolic plane, with infinite radius for all boundary circles. This amounts to requiring that the boundary circles are internally tangent to the unit circle. Theorem 6.3 implies that the locally univalent circle packing that we would obtain upon convergence is essentially unique. Since Theorem 6.5

states that there exists a univalent circle packing satisfying such boundary conditions, we know that the circle packing the algorithm is converging to is not only locally univalent, but also globally univalent.

We stop the Thurston algorithm short of convergence, as soon as the LPD condition is satisfied (or the GLPD condition, in case the graph is not a triangulation but a general 3-connected planar graph). The following lemma proves the correctness of this approach:

Lemma 7.1 *Conditions* LPD *and* GLPD *are open conditions in the neighborhood of circle packings in the sense that for all G and limit circle packing Φ_G, there exist a distance $\epsilon > 0$ such that for all circle mapping function ϕ, we have $d(\phi, \Phi_G) < \epsilon \Rightarrow \forall v \in V,\ \mathrm{LPD}(v, \phi)$ if G is a triangulation, and $d(\phi, \Phi_G) < \epsilon \Rightarrow \forall v \in V,\ \mathrm{GLPD}(v, \phi)$ if G is a 3-connected planar graph.*

Proof Taking Theorems 6.7 and 6.8 into account, it is enough to see that, in the case of circle packings, two neighbor circles have a common power diagram edge of positive length, and that the corresponding embedding of the centers is always strictly convex. □

7.3 Locality

Let us now examiine the locality of the computations involved in the algorithm. The Thurston algorithm requires each node of the triangulation to know the radii associated with its neighbors in order to update its own radius. This is the most local level of communication possible. We call it *G-locality.* In the case of 3-connected planar graphs, the Thurston algorithm needs to be generalized to require each vertex to know the radii of the vertices it shares a *face* with. This level of communication, which is slightly less local, is called *Gface-locality.*

The Thurston algorithm generates a set of radii, but in order to check the LPD or GLPD conditions, we need an actual embedding of the node and its neighbors. Such a layout of circles may be obtained by positioning the circles in a breadth-first order: once two neighbor vertices have their positions set, all other positions can be computed in this order. As for the computation of radii, this step is G-local in the case of a triangulation, but Gface-local in the case of 3-connected planar graphs. Similarly, one can see that checking LPD is G-local, whereas checking GLPD is Gface-local.

7.4 Experimental Results

We have implemented a simulation of this algorithm in MATLAB and tested it on random triangulations and 3-connected planar graphs containing around 50 vertices each, generated by E. Fusy's software [6]. We obtained greedy embeddings after a few hundred iterations (in general, less than 100 for triangulations, and between 100 and 500 for general 3-connected graphs). If we define an exact packing as a circle packing such that circles which should be tangent are indeed tangent, with an error on the distance between their centers within 1% of the smallest of the two radii, we can compare the number of iterations required to obtain a greedy power embedding with the number of iterations needed to obtain an exact packing: in the case of triangle graphs, we needed, on the average, a factor of 3.8 less iterations. In the case of general 3-connected planar graphs, we needed, on the average, a factor of 1.8 less iterations. Figures **??**, **??**, **??**, and **??** show two intermediary steps, the greedy power embedding and coin-graph embedding generated for the same input graph.

We did not implement the heuristic acceleration schemes proposed by Collins and Stephenson [4] because these heuristics rely on the global evaluation of the so-called *error reduction factor*. It would however be interesting to check whether a much more local evaluation of this parameter could still speed up the process significantly.

8 Conclusion and Future Work

We have described a modification of the Thurston algorithm for generating coin-graph embeddings, so that it is able to generate the embeddings required to support greedy power routing on a sensor network.

The algorithm is simple and local, thus may easily be implemented in a distributed manner on the sensor network.

Our current implementation uses a breadth-first traversal to locally compute the position of a vertex at each iteration once the radii have been adjusted. This involves simple and local computations, but may accumulate error in large networks. An optimized layout process that would spread the error evenly among the vertices could improve our results by triggering the termination conditions earlier. One way to do this is using the triangle layout method of ABF++ (Angle Based Flattening) [14], which involves solving a linear system for the vertex coordinates. Since this type of computation may be distributed among the vertices, it is a promising direction for future research. Alternatively, it might be possible to devise a way of checking LPD or GLPD from the radii only, without explicitly computing the vertex positions.

All algorithms for greedy routing rely on the input being a planar 3-connected graph, which is not very realistic. The simplest remedy is to extract a spanning subgraph of this type from the input and embed this. It is easy to see that adding back the non-planar edges after the embedding process does not harm the greedyness of the embedding. However, extracting such a subgraph is in itself a difficult problem. Thus an important problem is to devise a greedy embedding algorithm for general graphs.

References

[1] M. Ben-Chen, C. Gotsman and S. J. Gortler. Routing with Guaranteed Delivery on Virtual Coordinates. *Proc. Canadian Conf. on Comp. Geom, 2006.*

[2] A. Bobenko and B. Springborn. Variational Principles for Circle Patterns and Koebe's Theorem. *Trans. Amer. Math. Soc. 356 (2004), 659-689.*

[3] Y. Colin de Verdière. Empilements de cercles: Convergence d'une méthode de point fixe. *Forum Mathematicum, 1 (1989), 395-402.*

[4] C. Collins and K. Stephenson. A Circle Packing Algorithm. *Computational Geometry: Theory and Applications, 25 (2003), pp. 233-256.*

[5] M. B. Dillencourt and W. D. Smith. Graph-Theoretical Conditions for Inscribability and Delaunay Realizability. *Discrete Mathematics, 161(1-3):63-77, 1996.*

[6] É. Fusy. Quadratic Exact-size and Linear Approximate-size Random Sampling of Planar Graphs. *Proc. Analysis of Algorithms, 2005.*

[7] T. Hiroshima, Y. Miyamoto and K. Sugihara. Another Proof of Polynomial-Time Recognizability of Delaunay Graphs. *IEICE Trans. Fund., E83-A(4):627-638, 2000.*

[8] C. D. Hodgson, I. Rivin and W. D. Smith. A Characterization of Convex Hyperbolic Polyhedra and of Convex Polyhedra Inscribed in the Sphere. *Bull. of the Amer. Math. Soc. 27(3), 1992.*

[9] L. Kharevych, P. Schroeder and B. Springborn. Discrete Conformal Mappings via Circle Patterns. *Preprint, 2005.*

[10] E. Kranakis, H. Singh and J. Urrutia. Compass Routing on Geometric Networks. *Proc. CCCG 1999, pp. 51-54.*

[11] P. Morin. Online Routing in Geometric Graphs. *PhD thesis, Computer Science, Carleton University, 2001.*

[12] C. Papadimitriou and D. Ratajczak. On a Conjecture Related to Geometric Routing. *Proc. ALGOSENSORS 2004, pp. 9-17.*

[13] A. Rao, C. Papadimitriou, S. Shenker and I. Stoica. Geographic routing without location information. *Proc. MobiCom, 2003*

[14] A. Sheffer, B. Lévy, M. Mogilnitsky and A. Bogomyakov. ABF++ : Fast and Robust Angle Based Flattening. *ACM Transactions on Graphics, 24(2), 311-330 2005.*

[15] K. Stephenson. Introduction to Circle Packing. The Theory of Discrete Analytic Functions. *Cambridge University Press, April 2005.*

[16] W. Thurston. The finite Riemann Mapping Theorem. *Invited talk, An International Symposium at Purdue University on the occasion of the proof of the Bieberbach conjecture, March 1985.*

[17] W. Thurston. The Geometry and Topology of 3-Manifolds. *Princeton University Notes, Preprint.*