ARTICLE IN PRESS

Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Distributed computation of virtual coordinates for greedy routing in sensor networks

Mirela Ben Chen [a], Steven J. Gortler [b], Craig Gotsman [a,*], Camille Wormser [c]

[a] *Technion, Haifa, Israel*
[b] *Harvard University, United States*
[c] *INRIA, France*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Sensor network
Routing
Graph embedding

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 called "greedy routing" 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 the 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.

This paper is a combined and extended version of the preliminary results presented in [3,4].

* Corresponding author. Fax: +972 4 8295538.
E-mail addresses: mirela@cs.technion.ac.il (M. Ben Chen), sjg@cs.harvard.edu (S.J. Gortler), gotsman@cs.technion.ac.il (C. Gotsman), cwormser@sophia.inria.fr (C. Wormser).

0166-218X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2010.10.016

2. Previous work

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. An example of greedy routing is greedy Euclidean routing, which is based on Euclidean distance to the destination, or compass routing, based on angular distance to the destination [17]. An important question is the design of proximity (i.e. closeness) measures that guarantee the delivery of all packets, irrespective of the source or destination node. Since this measure is usually a distance in some space where the nodes have been embedded, the problem of positioning the nodes in such a space is referred to as the problem of computing a greedy embedding of a given network.

The most natural example of greedy routing is greedy Euclidean routing, where the proximity of nodes is measured simply by the Euclidean distance. This scenario has been studied in detail by Papadimitriou and Ratajczak [19], who conjectured that any 3-connected planar graph admits a greedy Euclidean embedding, namely, a greedy embedding for the Euclidean distance. An easy example is the subset of Delaunay-realizable triangle graphs, since it is easy to see that Delaunay triangulations are greedy Euclidean embeddings of their underlying graph.

While not able to prove their conjecture, Papadimitriou and Ratajczak propose other greedy routing schemes, most notably, 3D polyhedral routing. This consists of embedding the 3-connected planar graph as a polyhedron edge-tangent to the unit sphere in $\mathbb{R}^3$. A packet is routed by forwarding it to the neighbor vertex that maximizes the dot-product with the destination vertex. Such an embedding always exists, and they prove that the routing scheme always delivers.

Recently, Dhandapani [10] proved the conjecture of Papadimitriou and Ratajczak for the special case of a triangle graph. Using Schnyder embeddings of triangulations of the sphere, Dhandapani showed the existence of a greedy Euclidean embedding for any such triangulation. Unfortunately, the proof was not constructive.

Finally, the conjecture was proved by Leighton and Moitra [18]. See also the work of Angelini et al. [1] for a similar approach.

Other spaces have been considered as embedding spaces for the greedy routing problem. Kleinberg [15] studied the question of embedding the network in the hyperbolic plane, and was able to construct a greedy embedding in the hyperbolic plane for every connected finite graph. Note that the graph is neither assumed to be planar, nor to have any particular connectivity property, beyond connectedness. For these reasons, Kleinberg's results are particularly valuable in practical implementations. Furthermore, a distributed algorithm is presented, allowing the network to compute its own embedding, at the expense of a few broadcasting operations.

If longer labels per vertex may be tolerated, Flury et al. [11] show how to compute a greedy embedding of combinatorial unit disk graphs in $O(\log n)$ dimensions with bounded stretch.

The method of Sarkar et al. [20] is somewhat related to ours. They embed a planar triangulation in the plane using a so-called Ricci flow approximation to a conformal map, and the circle packings we compute are a classical discrete approximation to conformal maps. Alas, as Sarkar et al. describe in Section 3.3 of their paper, their embedding does not permit strictly greedy routing, and some cases they must resort to routing along edges using "virtual nodes".

3. Contribution

In this paper we present a greedy routing scheme for planar 3-connected graphs. The embedding is in $\mathbb{R}^2$, but the proximity measure used is not Euclidean. We show the relationship between our embedding and classical circle packings, and show how to modify Thurston's iterative algorithm for computing circle packings to compute our embeddings in a distributed manner.

Our greedy routing scheme is described in Section 4 and its relation to greedy polyhedral routing is investigated in Section 4.4. Section 5 reviews the notion of circle packings and Thurston's algorithm to compute them, while the design of suitable termination conditions for the Thurston algorithm is studied in Section 6. Section 7 is devoted to the description of the algorithm, before a final discussion on our validation experiments and future work in Section 8.

4. Greedy power routing

4.1. Power diagrams

Our greedy embedding scheme is intimately related to the classical power diagram. Let us first recall some definitions about (planar) power diagrams.

Definition 4.1. The *power* of a point x relative to a circle σ having center c and radius r is the real number

$$\mathrm{Pow}(x, \sigma) = \|x - c\|^2 - r^2.$$

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

More generally, the *power* of a circle τ having center c_τ and radius r_τ relative to a circle σ having center c_σ and radius r_σ is the real number

$$\mathrm{Pow}(\tau, \sigma) = \|c_\tau - c_\sigma\|^2 - r_\tau^2 - r_\sigma^2.$$

Note that with this definition, Pow is symmetric.

Let $\mathcal{C} = \{\sigma_1, \ldots, \sigma_n\}$ be a set of circles in the plane with centers c_i and radii r_i. $\mathrm{Pow}(x, \sigma_i) = \|x - c_i\|^2 - r_i^2$ is the power of x to σ_i, and $s_i = \|c_i\|^2 - r_i^2$ the power of the origin. To each σ_i, we associate the *power region* $L(\sigma_i)$ consisting of the points in the plane whose power relative to σ_i is no larger than their power relative to the other circles of $\mathcal{C}$.

Definition 4.2. The *power diagram* of $\mathcal{C} = \{\sigma_1, \ldots, \sigma_n\}$, denoted $L(\mathcal{C})$, is the cell complex whose cells are the power regions $L(\sigma_i)$ and their faces (see Fig. 2 for an example).

It is easy to verify that all power cells are convex and if all circles have the same radius, their power diagram is identical to the Voronoi diagram of their centers.

Thus power diagrams are a generalization of Voronoi diagrams. Along with this generalization come some nice properties, such as the fact that any affine diagram, that is, any diagram with affine edges between regions (and satisfying the incidence conditions required by the fact that it is a minimization diagram), is a power diagram, in the sense that there exists a set of circles whose power diagram is exactly the given affine diagram [2].

Power diagrams can be defined as the orthogonal duals of planar graphs embedded in the plane. Aurenhammer [2] proved that any orthogonal dual of a straight line plane graph is a power diagram of the vertices of the graph (with appropriate circles centered at the vertices). If one of these orthogonal duals is obtained with all radii equal, the planar graph is a Delaunay graph.

The set of power diagrams is also equivalent to the set of *regular embeddings*, i.e. the set of drawings that can be obtained as the projection of the edge structure of a convex polytope in $\mathbb{R}^3$ to the plane. This is a generalization of the fact that Voronoi diagrams are obtained as the projection of the edge structure of convex polytopes which are face-tangent to the unit paraboloid. In the dual setting, Delaunay triangulations are equivalent to projections to the plane of the convex hull of a set of points on the unit paraboloid (in $\mathbb{R}^3$) or to stereographic projections to the plane of the convex hull of a set of points on the unit sphere (in $\mathbb{R}^3$).

4.2. Power routing

The routing algorithm is greedy routing where the nodes are embedded as circles in the plane, and the circle power functions are used as distance functions. Namely, to route to destination t when at vertex v, forward to the neighboring vertex u such that $u = \mathrm{argmin}_{w \in N(v)} \mathrm{Pow}(w, t)$, where $N(v)$ is the set of neighbors of v.

In general, this greedy routing algorithm is not guaranteed to deliver. However, the freedom to choose the radius of each circle gives us some flexibility beyond the usual Euclidean distance so that the embedding can be made greedy.

4.3. Contained power diagrams

An orthogonal dual of a convex tiling is a planar embedding of the graph dual to the tiling, such that primal–dual edge pairs lie on orthogonal lines. We consider the setting in which the faces dual to boundary vertices are unbounded, and the vertex dual to the outer face is not embedded. For a 3-connected planar graph, there may exist many orthogonal primal/dual embedding pairs. Here we will be interested in pairs with a special property.

Definition 4.3. A *contained embedding* of a 3-connected planar graph is an orthogonal primal/dual embedding pair, such that each primal vertex is strictly contained in its dual face.

Lemma 4.4. *Any* 3*-connected planar graph and its dual have a contained embedding.*

Proof. The celebrated *kissing disks* theorem of Koebe and Andre'ev [16] states that any 3-connected planar graph and its dual can be simultaneously embedded in the plane such that each face is a convex polygon with an inscribed circle whose center coincides with the vertex of the dual corresponding to the face, and such that edges are perpendicular to their dual edges. Such an embedding is by definition a contained embedding (see Fig. 1). □

Note that a contained embedding of a graph is not necessarily unique. For example, if the graph happens to be a Delaunay-realizable triangulation, then any Delaunay realization and its dual Voronoi diagram are also a contained embedding for that graph.

As we have seen in the previous section, such contained embeddings are contained power diagrams. In terms of power diagrams, we have the following definition:

Definition 4.5. A power diagram is said to be *contained* if each site is contained in its cell (see Fig. 2).

This key containment property is a sufficient condition for the greedy power routing to deliver. To state this result, we adopt the following notations: let $G(V, E)$ be a combinatorial triangulation. Assume that G is planar and denote by B its boundary, which is a cycle. In the following, we study a map $\phi : V \to D^2 \times \mathbb{R}$, 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.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

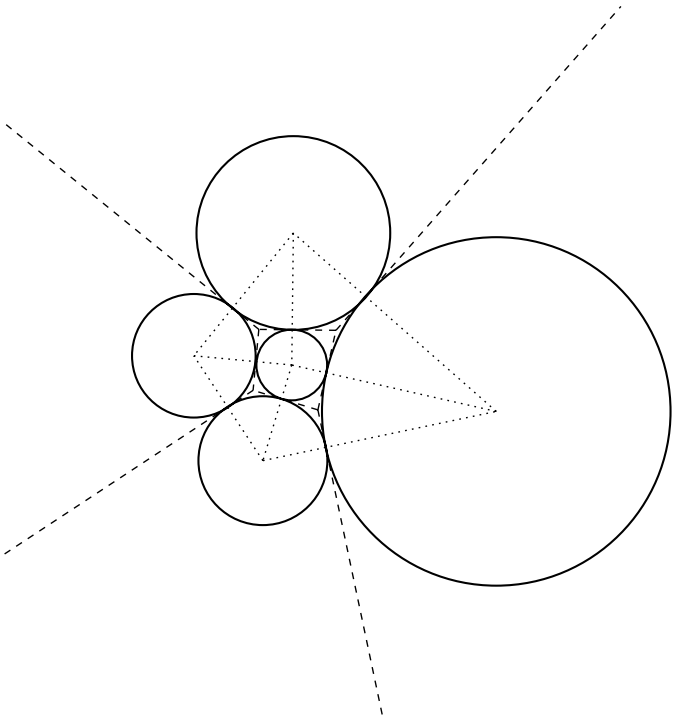

Fig. 1. A contained embedding obtained from the *kissing disks* theorem of Koebe and Andre'ev.

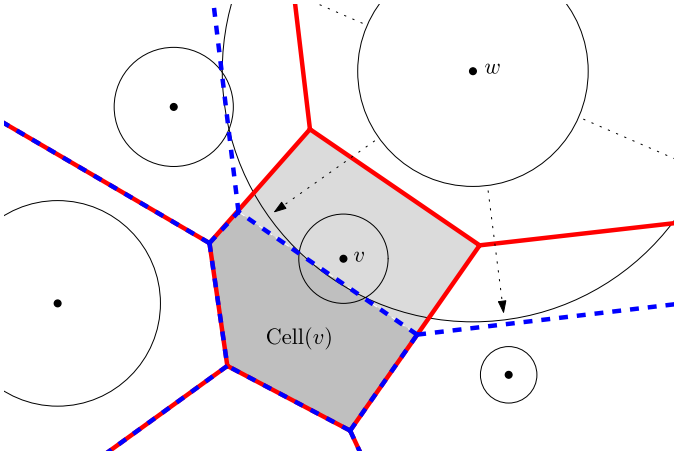

Fig. 2. As the radius of the circle around w grows, Cell(w) grows and Cell(v) shrinks. The power diagram becomes uncontained when v is no longer in Cell(v).

Theorem 4.6. *If the restriction of the power diagram of $\phi(V)$ to* 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. First note that in the special case that the embedding is a Delaunay triangulation, then all the radii are equal and greedy power routing is the same as greedy Euclidean routing.

In the general case, we must show that given a destination vertex t, each vertex v has a neighbor u in G such that $\text{Pow}(u, t) < \text{Pow}(v, t)$. This may be shown using an argument similar to that of Bose et al. [6] that the Delaunay triangulation is greedy. Specifically, consider the power diagram of the primal vertices with the given radii. Let e be the first edge of the power diagram which the line $v \to t$ intersects. There must exist such an edge, because in a contained embedding each vertex is strictly contained in its dual face, so v and t must lie in different cells of the power diagram. Let u be the vertex whose cell is adjacent to v's cell through e, and l be the line supporting e. Edge e is part of the restricted power diagram. Since the adjacency graph of the restricted power diagram is a subgraph of G, u is a neighbor of v in G.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

Every point x on l is equidistant from u and v : $\mathrm{Pow}(x, v) = \mathrm{Pow}(x, u)$. Every point y in the half plane created by l that contains u is closer to u than to v: $\mathrm{Pow}(y, u) < \mathrm{Pow}(y, v)$. By the definition of u, t lies in the half plane which is closer to u, hence $\mathrm{Pow}(t, u) < \mathrm{Pow}(t, v)$. It remains to show that the routing terminates at the destination vertex t. But, by construction, every vertex is strictly contained in its dual cell, hence all vertices $v \neq t$ in the embedding satisfy $\mathrm{Pow}(t, t) < \mathrm{Pow}(t, v)$. Thus, $\mathrm{Pow}(\cdot, t)$ has a global minimum at t. This concludes the proof. □

4.4. Equivalence to polyhedral routing

Before going deeper into the study of greedy power routing, we first show the equivalence between greedy power routing and greedy polyhedral routing, as described by Papadimitriou and Ratajczak [19].

Definition 4.7. *Greedy polyhedral routing* is greedy routing among the vertices of a convex polyhedron in $\mathbb{R}^3$ containing the origin O, by greedily maximizing $\langle Ov, Ot\rangle$ where v is the current vertex, and t the destination vertex.

We use elementary geometric arguments, but rely on what is simply the projective equivalence of polarities with respect to the paraboloid and with respect to the sphere.

4.4.1. Polarity

Denote by $\mathbb{S}^2$ the unit sphere of $\mathbb{R}^3$, and by O its center. The symbol $\langle ., .\rangle$ denotes inner product of two vectors.

Definition 4.8. The *polar hyperplane* of a point P different from O, denoted $\pi(P)$, is the plane defined by the equation $\langle \vec{OP}, x\rangle = 1$. The point P is called its *polar point*. We denote by $\mathcal{C}(P)$ the intersection $\pi(P) \cap \mathbb{S}^2$, and by $\mathrm{pr}_P(Q)$ the oriented distance between O and the projection of Q on (OP): $\mathrm{pr}_P(Q) = \langle OP, OQ\rangle / \sqrt{\langle OP, OP\rangle}$.

In other words, if P is outside $\mathbb{S}^2$, the circle $\mathcal{C}(P)$ is the locus of points x such that (Px) is tangent to $\mathbb{S}^2$, and $\pi(P)$ is the plane containing $\mathcal{C}(P)$. Note that, by definition, $\pi(P)$ is orthogonal to (OP). Let us now recall the following lemma.

Lemma 4.9. *For any two points P and Q outside $\mathbb{S}^2$,*

$$P \in \pi(Q) \Leftrightarrow Q \in \pi(P) \Leftrightarrow \mathcal{C}(P) \perp \mathcal{C}(Q).$$

Proof. In the following, we use the notations $p = \vec{OP}$ and $q = \vec{OQ}$. We have the following equivalences:

$$P \in \pi(Q) \Leftrightarrow \langle q, p\rangle = 1 \Leftrightarrow Q \in \pi(P).$$

Furthermore, the tangent vectors to $\mathcal{C}(P)$ and $\mathcal{C}(Q)$ at an intersection point x are collinear to $p \times x$ and $q \times x$. The scalar product of these vectors is

$$\langle p \times x, q \times x\rangle = \langle (q \times x) \times p, x\rangle = \langle q, p\rangle\langle x, x\rangle - \langle x, p\rangle\langle x, q\rangle = 1 \cdot 1 - 1 \cdot 1 = 0.$$

This concludes the proof. □

It easily follows from this lemma that the oriented angle of intersection $\alpha(P, Q)$ of two circles $\mathcal{C}(P)$ and $\mathcal{C}(Q)$ ($0°$ in the case of tangency) depends only on the distance $\mathrm{pr}_P(Q)$ between O and the projection of Q on (OP), and is a locally increasing function of this parameter. If we restrict Q so that $\mathcal{C}(Q)$ does not contain P we obtain an increasing function.

4.4.2. Stereographic projection

Recall that the stereographic projection and its inverse $\psi : \mathbb{R}^2 \to \mathbb{S}^2$ map circles to circles and preserve the angles of intersection between circles.

As in the previous section, we denote by $\mathrm{pr}_P(Q)$ the distance between O and the projection of Q on (OP). Let D, C_1 and C_2 be three circles in the plane, such that $\mathrm{Pow}(D, C_1) < \mathrm{Pow}(D, C_2)$. This inequality is independent of the radius of D. Thus, by adapting this radius, we may assume that C_1 and D intersect. Let us first assume that C_2 intersects D too. In this case, the angles of intersection satisfy $\alpha(D, C_1) > \alpha(D, C_2)$. Denote by P, Q_1 and Q_2 the points such that $\psi(D) = \mathcal{C}(P)$, $\psi(C_1) = \mathcal{C}(Q_1)$ and $\psi(C_2) = \mathcal{C}(Q_2)$. Since ψ preserves the angles of intersection, we have $\alpha(P, Q_1) > \alpha(P, Q_2)$. The previous section then implies that $\mathrm{pr}_P(Q_1) > \mathrm{pr}_P(Q_2)$. If C_2 does not intersect D, considering a second larger circle with the same center as D and orthogonal to C_2 provides the same conclusion.

This fact can be summarized as follows:

Lemma 4.10. *If $\mathcal{X}$ is a set of circles, and Y is another circle such that no circle of $\mathcal{X}$ contains the center of Y, then for any point P the extrema*

$$\min_{X \in \mathcal{X}} \mathrm{Pow}(P, X) \quad \textit{and} \quad \max_{X \in \mathcal{X}} \mathrm{pr}_P(\mathcal{C}^{-1}(\psi(X)))$$

are obtained at the same $X_0 \in \mathcal{X}$.

Note that these two quantities are mapped to each other by a homography. This explains why a restriction is needed in order to have a monotonic function.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

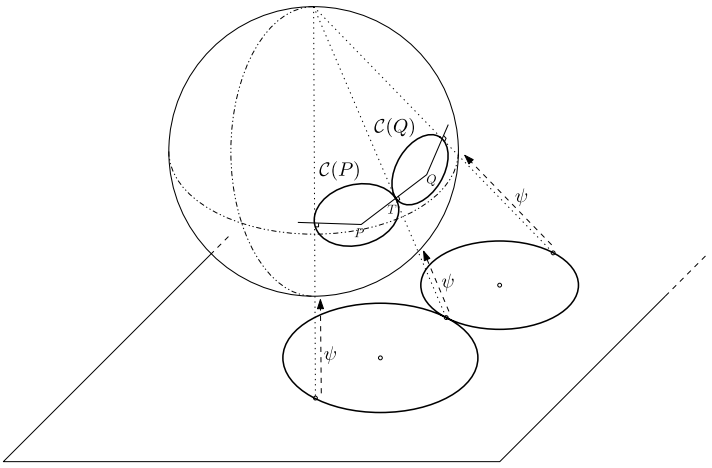

Fig. 3. Two circles in the plane and their images (P and Q) on the sphere through $\mathcal{C}^{-1} \circ \psi$.

4.4.3. Routing equivalence

Given a set of circles $\mathcal{X}$ such that no circle contains the center of another circle, Lemma 4.10 shows that greedy polyhedral routing on $\mathcal{C}^{-1}(\psi(X))$ (see Definition 4.7) generates exactly the same paths as greedy power routing on $\mathcal{X}$.

It follows that any set of circles on which greedy power routing delivers, composed with the mapping $\mathcal{C}^{-1} \circ \psi$, provides a polyhedron on which greedy polyhedral routing delivers. Furthermore, the following lemma relates the equivalent special cases which interest us:

Lemma 4.11. *The transformation $\mathcal{C}^{-1} \circ \psi$ maps a circle packing to a polyhedron edge-tangent to $\mathbb{S}^2$ (see Fig. 3).*

Proof. ψ maps tangent circles in the plane to tangent circles on the sphere. Denote by $\mathcal{C}(P)$ and $\mathcal{C}(Q)$ two such tangent circles on $\mathbb{S}^2$, with P and Q being their polar points, and denote by T their tangency point. By construction, (PT) is tangent to $\mathbb{S}^2$ at T and orthogonal to $\mathcal{C}(P)$ at T. Similarly, (QT) is tangent to the sphere at T and orthogonal to $\mathcal{C}(Q)$ at T. Hence, both lines (PT) and (QT) belong to the tangent plane of $\mathbb{S}^2$ at point T, and both are orthogonal to the common tangent of $\mathcal{C}(P)$ and $\mathcal{C}(Q)$ at T. It follows that P, T and Q are co-linear. This proves that the segment linking the images of two tangent circles by $\mathcal{C}^{-1} \circ \psi$ is tangent to $\mathbb{S}^2$. The result follows. □

One can also show the connection between the containment property of power diagrams, and the property that Papadimitriou and Ratajczak proved to be a sufficient condition for greedy polyhedral routing to deliver:

Definition 4.12. Let P be a convex polyhedron in $\mathbb{R}^3$ containing the origin O. A *supporting hyperplane* of P at vertex v of P is a hyperplane that contains v but does not intersect P otherwise.

A polyhedron P is said to have *orthogonal support* if for each vertex v of P, the plane orthogonal to (Ov) and containing v is a supporting hyperplane.

Papadimitriou and Ratajczak proved that having orthogonal support is a sufficient condition for a polyhedron to provide greedy routing that delivers.

Lemma 4.13. *A set of circles defines a contained power diagram if and only if its image by $\mathcal{C}^{-1} \circ \psi$ is a polyhedron P with orthogonal support.*

Proof. Denote by v and w two vertices of a convex polyhedron P. Circles $C_v = C(c_v, r_v) = \psi^{-1} \circ \mathcal{C}(v)$ and $C_w = C(c_w, r_w) = \psi^{-1} \circ \mathcal{C}(w)$ are the corresponding circles in the plane. Denote by h_v the plane orthogonal to (Ov) and containing v. Then w belongs to h_v if and only if the radical axis of C_v and C_w (i.e. the power diagram bisector, which is, in this case, the line passing through the intersection points of the two circles) passes through c_v. In other words, w belongs to h_v if and only if c_v belongs to the boundary of the power region of C_w in the power diagram of $\{C_v, C_w\}$.

The result then follows from Lemma 4.10. □

This completes the parallel between the two routing schemes. This parallel implies that the algorithm we design in the following sections allows the computation of more general greedy polyhedral embeddings than those edge-tangent to the sphere, as proposed by Papadimitriou and Ratajczak.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

5. Circle packing

5.1. Definitions

As we have seen in Lemma 4.4, kissing disks, also called circle packings, are an example of contained power diagrams, and, as such, are a greedy power embedding of their tangency graph. More formally:

Definition 5.1. 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 5.2. 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.

We now state a few important results about circle packings. A detailed presentation of the subject can be found in [22].

Theorem 5.3 ([22], p. 18)**.** *Given a planar triangulation $G(V, E)$, and 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 values as their radii.*

Essentially unique is to be understood as up to isometry.

Definition 5.4. A G-circle packing is said to be *univalent* if its circles have mutually disjoint interiors.

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

Theorem 5.5 ([22], p. 62)**.** *Let G be a combinatorial closed disk (that is, simply connected, finite, triangulation). Then there exists an essentially unique univalent circle packing $\mathcal{P}_G$ contained in the unit disk such that any boundary circle is internally tangent to the unit disk.*

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

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 5.6. 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 $\gamma(w_i)$, $1 \leq i \leq n$.

5.2. Practical computations of circle packings

Various methods exist for the computation of circle packings. The oldest and simplest one, which we will study in detail and build upon, is the Thurston algorithm [23]. It is an iterative algorithm which greedily updates the radii of the circles until they converge to values compatible with circle packing. Various other algorithms have surfaced since the inception of the original Thurston algorithm. Before presenting the details of the Thurston algorithm, we briefly describe two other algorithms relevant to our study.

5.2.1. The Springborn–Bobenko algorithm

Springborn and Bobenko [5] have proposed a general framework for dealing with the so-called *circle patterns*, which are sets of circles with non-zero intersection angles instead of the simpler tangency condition of circle packings. They characterize the intersection angles for which such circle patterns exist, and then define convex functionals on circle patterns which are minimized when the required conditions on these intersection angles are satisfied.

These ideas have been applied by Kharevych et al. [14] to the conformal parametrization of discrete 3D surfaces. They show how to apply the variational characterization of circle patterns of [5] to the practical computation of circle patterns with prescribed intersection angles.

Applying these methods to the special case of circle packings is easy. However, the minimization procedure is not directly amenable to distribution among network nodes.

5.2.2. Discrete Ricci flow

Chow and Luo [7] have considered a completely different approach to the question of circle packing. They describe a discretization of Hamilton's Ricci flow and prove that it converges to a circle packing with prescribed adjacency relations. This implies an algorithm for computing circle packings, which is proved to converge exponentially fast.

While very efficient, this algorithm requires a periodic global rescaling of the circle radii, which prevents distribution of the computation among network nodes.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

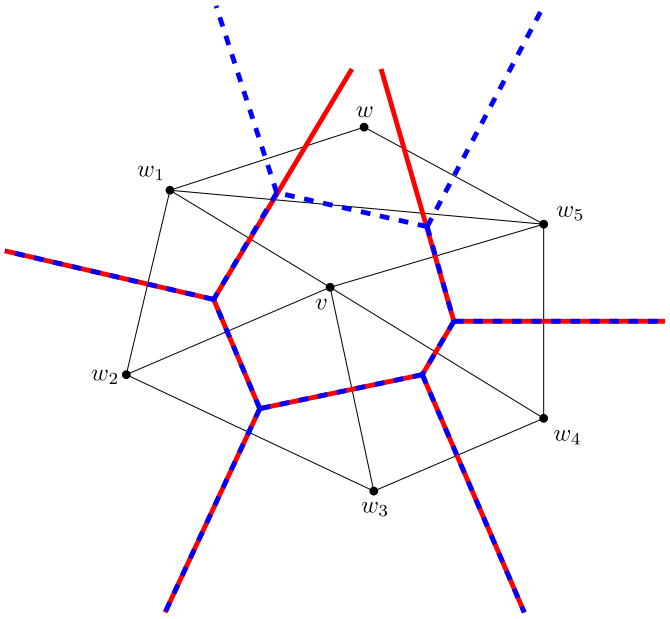

Fig. 4. The local cell $\mathrm{Cell}_G(v)$ (solid lines) contains the power diagram cell (dashed lines) and contains another vertex w.

5.2.3. The Thurston algorithm

In this section we present the algorithm that Thurston [23] designed for the numerical computation of circle packings.

The 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 [8]. See [9] for a practical and efficient implementation of this algorithm.

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 5.6.

In the following, we represent the Thurston algorithm by a sequence of the 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 isometry of the hyperbolic plane, namely, some Möbius transformation.

There are two reasons why we focus on Thurston's algorithm: it is an extremely simple algorithm, and it can be distributed in a straightforward manner. However, there is one drawback in this algorithm, beyond its relative slowness. It provides only an *approximation* of the desired circle packing. Computing the exact one would require an infinite number of steps.

In what follows, we show how to overcome this such that only a finite number of steps are required.

6. Local termination conditions

In order to stop the iterations of Thurston algorithm, we need a termination condition that would guarantee that the result is at least a contained power diagram, with the correct adjacency relations. This is sufficient to enable greedy power routing. We need to ensure, however, that the algorithm may be distributed, including checking the termination condition.

6.1. Triangulated case

Recall that we study a map $\phi : V \to D^2 \times \mathbb{R}$, which associates to each vertex v a point $p(v)$ in the unit disk and a radius $\sigma(v)$:

$$\phi = (p, \sigma).$$

The boundary of G is denoted by B.

Definition 6.1. 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 Fig. 4).

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).

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

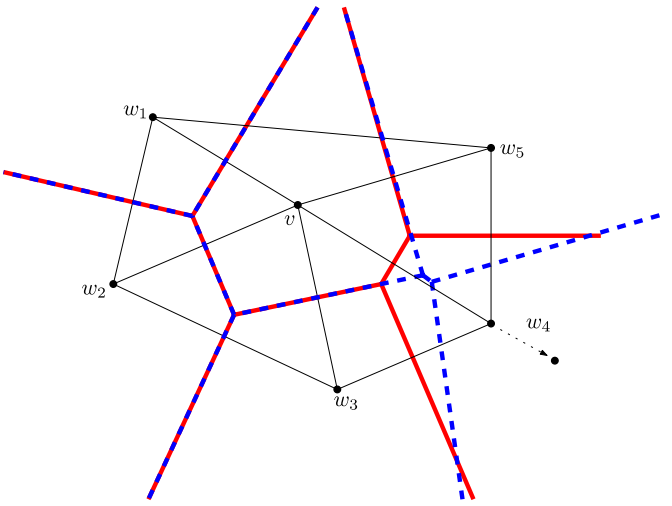

Fig. 5. As w_4 moves away from v, LPD(v, ϕ) becomes unsatisfied (while the solid lines diagram becomes the dashed lines diagram), because $\text{Cell}_G(v)$ and $\text{Cell}_G(w_4)$ are not adjacent anymore, whereas edge $[vw_4]$ exists in G.

Definition 6.2. For any vertex $v \in V$, we say that property LPD(v, ϕ) (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 $\text{Cell}_G(v)$ contains $p(v)$ and the cells adjacent to it are exactly those of $w_1, \ldots, w_n$ (in this order, see Fig. 5);
- 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 $\{\phi(v), \phi(w_1), \ldots, \phi(w_n)\}$, $\text{Cell}(v) \cap \text{Cell}(w_1) \cap \text{Cell}(w_n)$ is either empty (which means that $\text{Cell}(v)$ 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.

We are now ready to state the central theorem of this section:

Theorem 6.3. *If*

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

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

Proof. From now on, we denote by $\text{Cell}(w)$ the cell of $\phi(w)$ in the power diagram of $\phi(V)$, and by $\text{Cell}_G^v(w)$ 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 $\text{Conv}(p(V))$.

We now prove that $\rho(\text{Cell}_G^v(v)) = \rho(\text{Cell}(v))$ for all $v \in V$. First note that $\text{Cell}(v) \subset \text{Cell}_G^v(v)$ for all $v \in V$ implies that $\cup_{v \in V} \rho(\text{Cell}_G^v(v)) = \text{Conv}(p(V))$.

For each vertex $v \in V$, we consider the usual lifting to the polar hyperplane $\ell_v : x \mapsto (x, 2\langle x, \phi(v)\rangle - \|\phi(v)\|^2 + r(v)^2)$ in dimension 3. 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 $\ell_v(\rho(\text{Cell}_G^v(v)))$ can be glued into a convex terrain over the convex domain $\text{Conv}(p(V))$ (see Fig. 6).

If v and w are neighbors in G and $v \notin B$, let p and q be the two vertices opposite 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 LPD(v, ϕ) and LPD(w, ϕ) imply that the segment $[\alpha\beta]$ is an edge common to $\text{Cell}_G^v(v)$ and $\text{Cell}_G^w(w)$ because the four vertices v, w, p and q will all appear in the computations of the border of both cells.

This implies that $\ell_v(\text{Cell}_G^v(v))$ and $\ell_w(\text{Cell}_G^w(w))$ can be glued together along their common edge which is $[AB] = \ell_v([\alpha\beta]) = \ell_w([\alpha\beta])$. Furthermore, one can see that the angle between $\ell_v(\text{Cell}_G^v(v))$ and $\ell_v(\text{Cell}_G^w(w))$ along $[AB]$ is convex, because it is true for the local diagram of v and its neighbors.

Now consider the case where both v and w are boundary vertices. Let p be the vertex opposite (v, w) in G and consider the edge $e(v, w) = \text{Cell}_G^v(v) \cap \text{Cell}_G^v(w)$. Hypothesis 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. $e(v, w)$ is also perpendicular to the line $(p(v)p(w))$ and 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 $\ell_v(\text{Cell}_G^v(v))$ and $\ell_w(\text{Cell}_G^w(w))$ can be glued together along this convex edge.

Finally, we obtain that the $\ell_v(\text{Cell}_G^v(v))$ can be glued together into a locally convex polyhedral terrain $\mathcal{P}$ over the convex domain $\text{Conv}(p(V))$. It follows that $\mathcal{P}$ is globally convex and is in fact the restriction of a convex polytope and that the

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

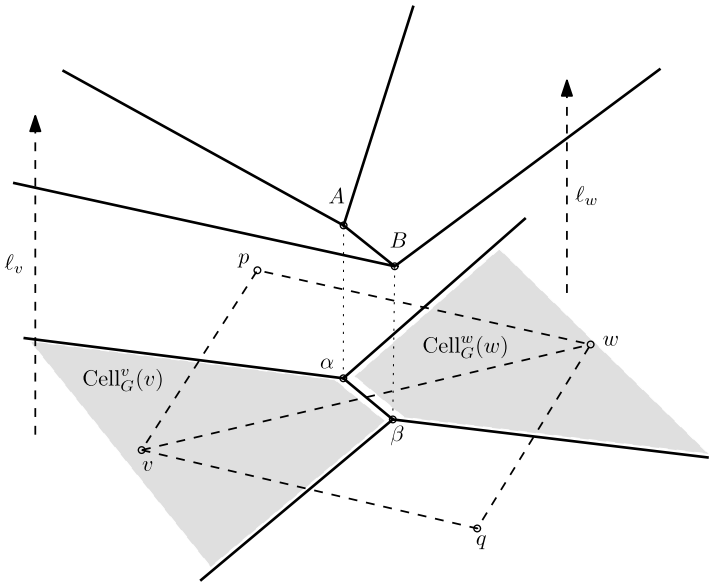

Fig. 6. Lifting two local cells that share an edge.

projection of its edges onto $\text{Conv}(p(V))$ 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.

Note that the sites on the boundary of G may not be in convex position. In particular, if the power diagram were not restricted to $\text{Conv}(p(V))$ (as we have seen in the proof, restricting to $\mathbb{D}^2$ is in fact sufficient), the cells of vertices which are not connected in G may be adjacent, creating an adjacency graph bigger than G. □

We can now state the following corollary of Theorems 4.6 and 6.3:

Corollary 6.4. *If*

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

then greedy power routing delivers on ϕ. □

6.2. Generalized Papadimitriou and Ratajczak result

Papadimitriou and Ratajczak [19] provided geometric conditions on embeddings of 3-connected planar graphs which characterize greedy Euclidean embeddings. We now present this result in the more general context of arbitrary distance functions, and explain how it relates to Section 6.1. We will need this for the extension of the results of Section 6.1 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
$$\text{Cell}(v) = \{x \in \mathbb{R}^2, d_v(x) \leq d_w(x), \ \forall w \in V\}$$
- the reciprocal one, where the cell of a site v, called the *reciprocal cell* is defined as
$$\text{Cell}^\circ(v) = \{x \in \mathbb{R}^2, d_x(v) \leq d_x(w), \ \forall w \in V\}.$$

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 $\text{Cell}_G(v)$ of a vertex v of an embedded graph G, we can define the local *reciprocal* cell $\text{Cell}_G^\circ(v)$ and state a generalized version of the characterization of Papadimitriou and Ratajczak.

Theorem 6.5. *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* $\text{Cell}_G^\circ(v)$ *contains no vertex other than* v.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

Proof. The proof is exactly the same as in Theorem 1 of [19]. □

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 $\text{Cell} = \text{Cell}^\circ$ and $\text{Cell}_G = \text{Cell}_G^\circ$. 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 generalize Theorem 6.5:

Theorem 6.6. *Greedy power routing delivers if and only if for each vertex $v \in V$, the local cell $\text{Cell}_G(v)$ for the power distance contains no vertex other than $p(v)$ (see Fig. 4).* □

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:

$$\begin{array}{ccc} & \text{Theorem 6.3} & \\ \forall v \in V, & \Longleftrightarrow & \forall v \in V, \\ \text{LPD}(v, \phi) & & v \in \text{Cell}_G(v) = \text{Cell}(v) \\ & & \text{Theorem 4.6} \quad \Downarrow \\ \text{Greedy routing} & \Longleftrightarrow & \forall v \in V, \\ \text{delivers on } \phi(G) & \text{Theorem 6.6} & \text{Cell}_G(v) \text{ contains only } v. \end{array}$$

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 $\phi(V)$". Theorem 6.3 proves the left-to-right implication, and the right-to-left one is easy to check.

6.3. Non-triangulated case

Let us now consider the more general case of a 3-connected planar graph. As in Section 6.1 for triangulated graphs, we present local sufficient conditions for greedy power routing to deliver on general 3-connected planar graphs. The locality of the conditions is discussed in Section 7.3.

In the previous section, we proved that satisfying LPD at every vertex implied that G is the adjacency graph of the power diagram of $\phi(V)$. This cannot be the case if G is not a triangulation: such a 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 the following result:

Lemma 6.7. *If a set of points $\{p_1, \ldots, p_n\}$ is in convex position, for any radii $(\sigma_i)_{1 \le i \le n}$, the adjacency graph of the power diagram of the circles $\mathcal{C}(p_i, \sigma_i)$ is a triangulation of* $\text{Conv}(\{p_1, \ldots, p_n\})$.

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

Definition 6.8. 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 6.7). The resulting triangulation of G is called the ϕ-triangulation of G and is denoted by $G(\phi)$ (see Fig. 7).

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 6.9. For any vertex $v \in V$, we say that property $\text{GLPD}(v, \phi)$ (Generalized Local Power Diagram) is satisfied if and only if the faces incident to v are convex, property $\text{LPD}(v, \phi)$ is satisfied in $G(\phi)$ and for each non-triangle face $f = (v, w_1, \ldots, w_n)$ incident to v, the local cell $\text{Cell}_G(v)$ of v in G intersects f only along segments $[w_n v]$ and $[v w_1]$ (see Fig. 8).

Note that, in the last condition, the local cell is computed in G, and not in $G(\phi)$ (see Definition 6.8): otherwise, the condition is trivially satisfied.

Theorem 6.10. *If p is a convex embedding and*

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

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

Proof. From the proof of Theorem 6.3, we know that $\text{LPD}(v, \phi)$ being satisfied for every vertex v implies that the local cell $\text{Cell}_{G(\phi)}(v)$ computed in $G(\phi)$ is exactly the cell of the power diagram of $\phi(V)$, and that this diagram is a contained embedding of $G(\phi)$.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

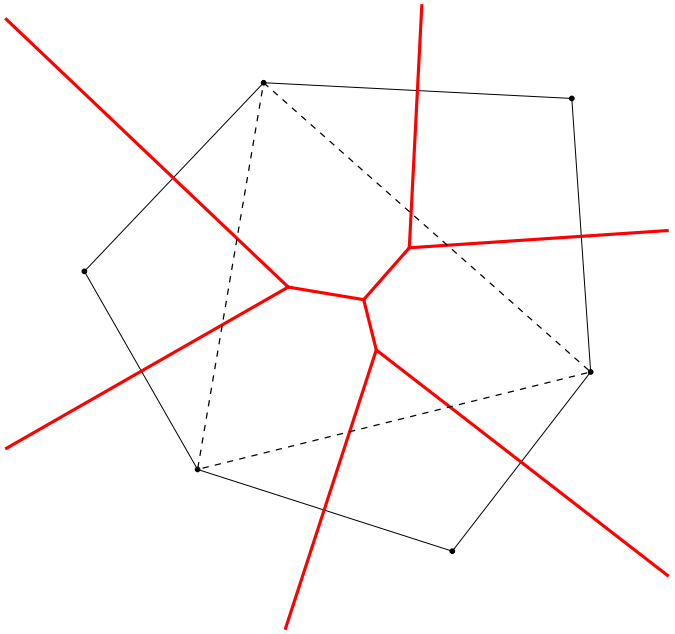

Fig. 7. A face with 6 vertices embedded by ϕ with the regular triangulation of its vertices: G (solid lines) is triangulated into $G(\phi)$ (solid and dashed lines).

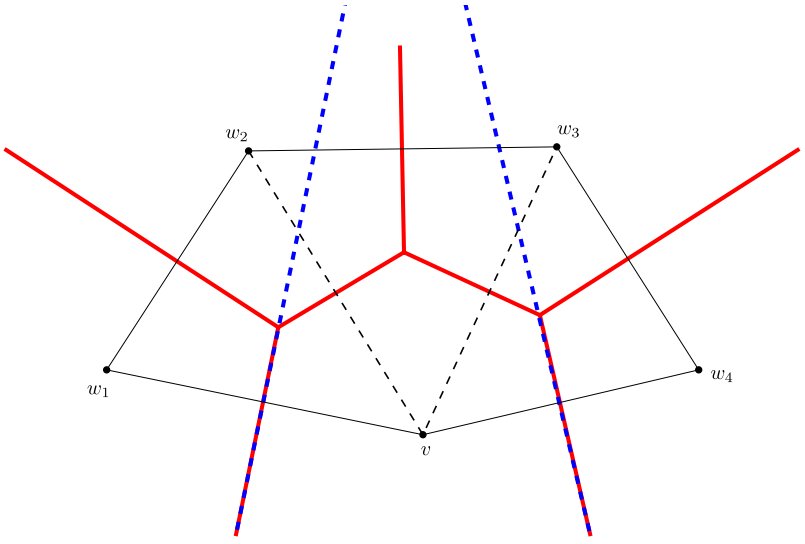

Fig. 8. A face (solid edges) with 5 vertices, with GLPD(v, ϕ) not satisfied: the local cell of v (dashed lines) crosses the boundary of the face not only on $[w_1v]$ and $[vw_4]$ but also on $[w_2w_3]$, which is forbidden.

We need the local cell $\text{Cell}_G(v)$ computed in G to be empty of other vertices. We know that $\text{Cell}_{G(\phi)}(v) \subset \text{Cell}_G(v)$. We now prove that the difference $\text{Cell}_G(v) \setminus \text{Cell}_{G(\phi)}(v)$ is contained in the union of the faces incident to v. Note that $\text{Cell}_{G(\phi)}(v)$ is not itself contained 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 $W = N_{G(\phi)}(v) \setminus N_G(v)$. Denote by $\text{Cell}_f(v)$ the cell of v in the power diagram of $\{v\} \cup N_G(v) \cup W_f$.

By convexity of f, and using the fact that the local cells of the w_i are not allowed to cross f along the segments $[w_nv]$ and $[vw_1]$, one can easily see that $\text{Cell}_G(v) \setminus \text{Cell}_f(v)$ is contained in f. Since $\text{Cell}_{G(\phi)}(v) = \cap_f \text{Cell}_f(v)$, 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 6.6 and 6.10:

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

Corollary 6.11. *If the first component p of ϕ is a convex embedding and if*

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

then greedy power routing delivers on ϕ. □

6.4. Relation between circle packings, LPD and GLPD

The following theorems show that the conditions that we have described are indeed satisfied by the limits of the Thurston algorithm, namely circle packings.

Theorem 6.12. *If G is a planar triangulation and if $\phi(G)$ is a G-circle packing of the unit disk, then*

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

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

Theorem 6.13. *If G is a 3-connected planar graph and if $\phi(G)$ is a G-circle packing of the unit disk, then*

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

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. □

7. Algorithms

7.1. Computing a greedy power embedding

We now derive from Sections 6.1 and 6.3 a distributed algorithm for the computation of a contained power diagram.

The algorithm consists simply of augmenting Thurston's iterative circle packing algorithm (see Section 5.2.3) with the conditions LPD (or GLPD) as termination conditions. Note that the Thurston algorithm itself has no concrete termination condition: it is an iterative process which is guaranteed to converge, and that in practice is run as many times as needed until some condition measuring convergence is met. Typically, some threshold on the angular error is used as a termination condition. However, it is not obvious that any such threshold on the angular error can guarantee that a contained power diagram is achieved.

The correctness of the algorithm follows from Section 6.4, since, in the worst case, the conditions LPD (or GLPD) will be satisfied when the algorithm converges to a circle packing, which is guaranteed. We now describe the algorithm and discuss its correctness.

7.2. Termination

Our algorithm consists of running the Thurston algorithm to compute a circle packing in the Poincaré model of the hyperbolic plane, initialized with infinite radii for all boundary circles. This amounts to requiring that the boundary circles are internally tangent to the unit circle. Theorem 5.3 implies that the locally univalent circle packing that we would obtain upon convergence is essentially unique. Since Theorem 5.5 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 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).

More precisely, the steps of the algorithm are as follows (with some integer parameter $N > 0$):

(1) set all boundary radii to infinity and all internal radii to 1;
(2) update all internal radii by applying N steps of Thurston's algorithm in the hyperbolic plane;
(3) fix the positions of two neighbor disks and sweep the network to compute the Euclidean layout ϕ of the circles in the Poincaré unit disk representation of the hyperbolic plane;
(4) if $\mathrm{LPD}(v, \phi)$ (or $\mathrm{GLPD}(v, \phi)$ in the non-triangulated case) is not satisfied for some v, go to step 2. Otherwise, return the current layout.

Note that in the non-triangulated case, steps 2, 3 and 4 will require the network to emulate a triangulation of the graph. Additionally, the network has to be able to detect the state at which $\mathrm{LPD}(v, \phi)$ is satisfied at *all* nodes (step 4), at which point the algorithm terminates. This is complicated by the fact that $\mathrm{LPD}(v, \phi)$ being satisfied does not imply that it will continue

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

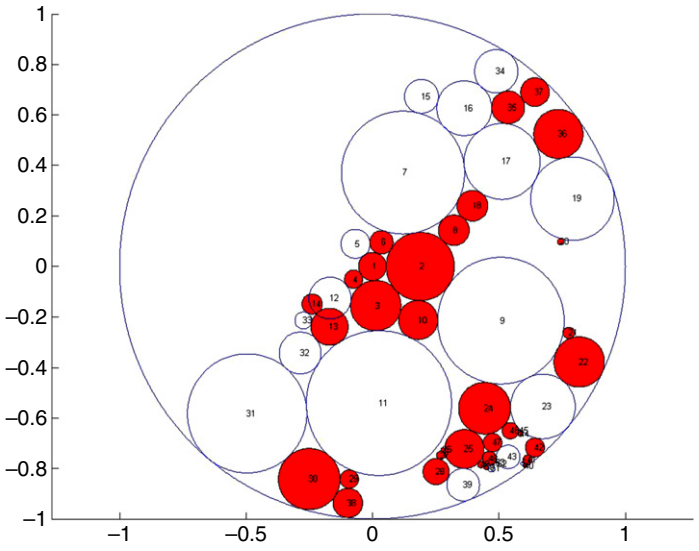

Fig. 9. After 6 iterations, the colored circles are the ones that already satisfy LPD.

to be satisfied at subsequent iterations (because of the activity at neighboring nodes). However, the following lemma proves that ultimately the algorithm will converge, namely, reach a state in which LPD(v, ϕ) is satisfied for all v. This state may then be detected by standard distributed algorithmic techniques.

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 exists a distance $\epsilon > 0$ such that for all circle mapping functions ϕ, we have $d(\phi, \Phi_G) < \epsilon \Rightarrow \forall v \in V$, LPD$(v, \phi)$ if G is a triangulation, and $d(\phi, \Phi_G) < \epsilon \Rightarrow \forall v \in V$, GLPD$(v, \phi)$ if G is a 3-connected planar graph.*

Proof. Using Theorems 6.12 and 6.13, it suffices to observe that, in the case of circle packings, two neighboring circles have a common power diagram edge of positive length, and that the corresponding embedding of the centers is always strictly convex, i.e. all the faces of the embedding are strictly convex. □

7.3. Locality

Let us now examine the locality of the computations involved in the algorithm. In the triangulated case, each node of the triangulation needs 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 a non-triangle 3-connected planar graph, each vertex needs to know the radii of the vertices it shares a *face* with. This level of communication, which is less local, is called *G-face-locality*.

The 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 triangular graph, but G-face-local in the case of 3-connected planar graphs. Similarly, one can see that checking LPD is G local, whereas checking GLPD is G-face-local.

8. Discussion

8.1. Experimental validation

We have implemented a simulation of the algorithm of Section 7 in MATLAB and tested it on random triangular graphs and 3-connected planar graphs containing around 50 vertices each, generated by Fusy's software [13]. We obtained greedy power 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. Figs. 9–12 show two intermediate steps, the greedy power embedding and the exact packing generated for the same input graph.

Note that the high non-uniformity of these random graphs, i.e. a short loop of edges may bound a region containing a large number of vertices (i.e. the graph contains small cuts), is a reason for the relatively low efficiency of the algorithm. This

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

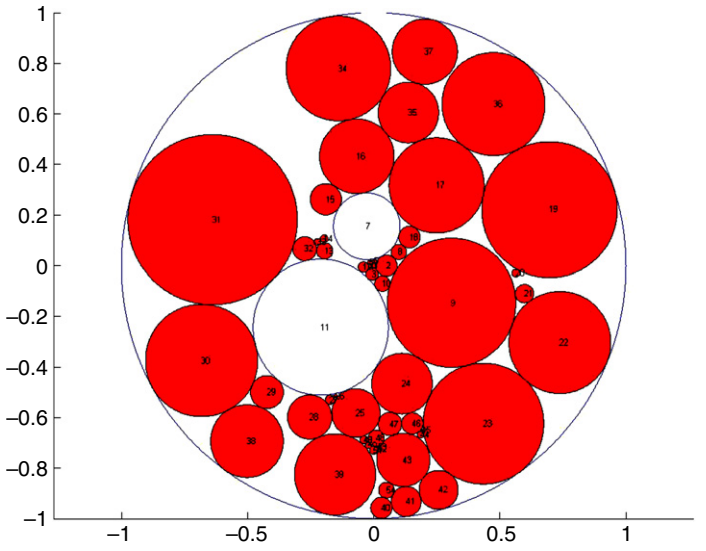

Fig. 10. After 29 iterations, only 2 circles still do not satisfy LPD.

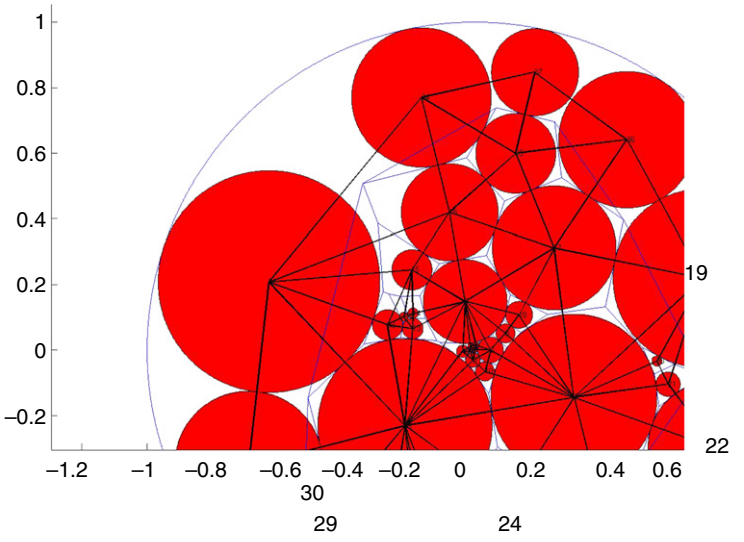

Fig. 11. After 32 iterations, LPD is satisfied everywhere: the embedding is a greedy power embedding.

kind of setting is not realistic in the case of sensor networks, where one would expect the planar graph to be a subgraph of a realistic communication graph such as a unit disk graph.

To test the scalability of the algorithm, we ran it on a set of random triangulations, generated by positioning n random points in the unit square, and computing their Delaunay triangulation. We then discarded the points' locations, and used the triangulation as input to the algorithm. For each triangulation, we computed the number of iterations until the algorithm converged, and the stretch ratio of the routing path for all pairs of vertices. Given two vertices s and t, the *stretch ratio* is defined as the ratio between the (graph-theoretic) length of the routing path from s to t and the (graph-theoretic) length of the shortest path between s and t. We repeated this experiment for triangulations with 50 to 500 vertices, generating 100 triangulations of each size. Fig. 13 shows the number of iterations until convergence, averaged over all triangulations of the same size. This indicates that the number of iterations is linear in the number of vertices. Figs. 14 and 15 show the mean and maximal stretch ratio respectively, averaged over all pairs of vertices, over all triangulations of the same size. For these size networks, the mean stretch seems to be bound by 1.25 and the maximal stretch by 7.0.

We did not implement the heuristic acceleration schemes proposed by Collins and Stephenson [9] 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.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

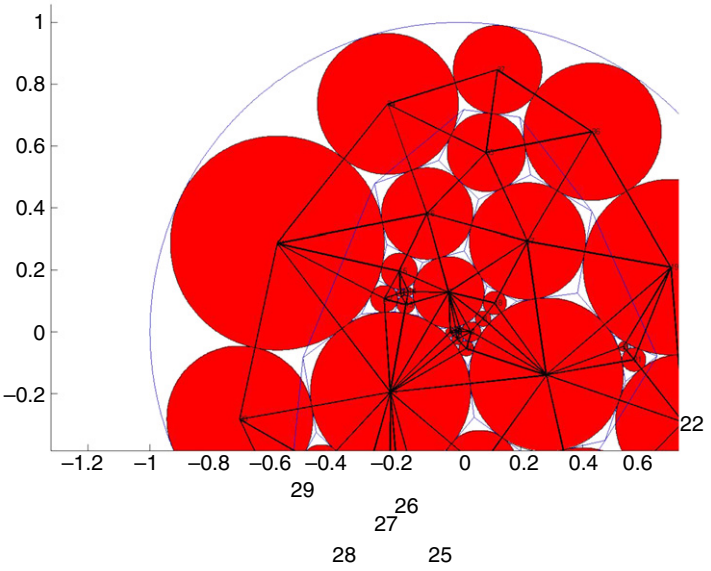

Fig. 12. After 128 iterations, the embedding is a circle packing.

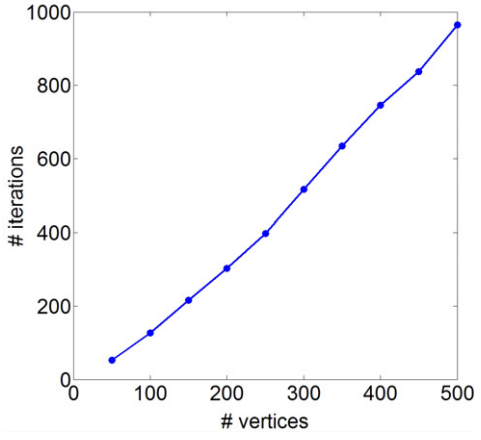

Fig. 13. Average number of iterations to convergence.

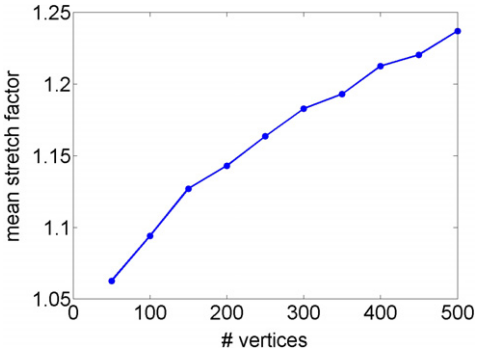

Fig. 14. Mean routing path stretch ratio.

8.2. Possible improvements

We have described a modification of the Thurston algorithm originally designed for generating circle packings, so that it is able to generate the embeddings required to support greedy power routing on a sensor network. The algorithm is simple

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016

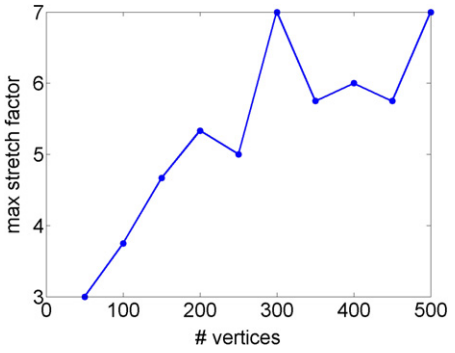

Fig. 15. Maximal routing path stretch ratio.

and G face-local, thus may easily be implemented in a distributed manner on the sensor network. However, our algorithm is not practical in case the domain contains big holes, which would function as large non-triangulated faces. A natural way of dealing with this problem would be to analyze the topology of the underlying domain and split it into simply connected parts which could be treated separately (see [12]).

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) [21], 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.

Most 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 a more general input and embed this. It is easy to see that adding back the non-planar edges after the embedding process does not harm the greediness of the embedding. However, extracting such a subgraph is in itself a difficult problem. Thus an important open problem is to devise a greedy embedding algorithm for general graphs.

References

[1] P. Angelini, F. Frati, L. Grilli, An algorithm to construct greedy drawings of triangulations, in: Proc. Graph Drawing 2008, Lecture Notes in Computer Science, vol. 5417, 2009, pp. 26–37.
[2] F. Aurenhammer, Power diagrams: properties, algorithms and applications, SIAM J. Comput. 16 (1987) 78–96.
[3] M. Ben Chen, C. Gotsman, S.J. Gortler, Routing with guaranteed delivery on virtual coordinates, in: Proc. Canadian Conf. on Comp. Geom., 2006.
[4] M. Ben Chen, C. Gotsman, C. Wormser, Distributed computation of virtual coordinates, in: Proc. Symp. Computational Geometry, 2007, pp. 210–219.
[5] A. Bobenko, B. Springborn, Variational principles for circle patterns and Koebe's theorem, Trans. Amer. Math. Soc. 356 (2004) 659–689.
[6] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, Wirel. Netw. 7 (6) (2001) 609–616.
[7] B. Chow, F. Luo, Combinatorial Ricci flows on surfaces, J. Differential Geometry 63 (1) (2003) 97–129.
[8] Y. Colin de Verdière, Empilements de cercles: convergence d'une méthode de point fixe, in: Forum Mathematicum, vol. 1, 1989, pp. 395–402.
[9] C. Collins, K. Stephenson, A circle packing algorithm, Comput. Geom., Theory Appl. 25 (2003) 233–256.
[10] R. Dhandapani, Greedy drawings of triangulations, in: Proc. Symp. Disc. Alg., 2008, pp. 102–111.
[11] R. Flury, S.V. Pemmaraju, R. Wattenhofer, Greedy routing with bounded stretch, in: Proc. INFOCOM, 2009.
[12] S. Funke, N. Milosavljevic, Network sketching or: "how much geometry hides in connectivity? — part II", in: Proc. Symp. on Disc. Alg., 2007.
[13] É. Fusy, Quadratic exact-size and linear approximate-size random sampling of planar graphs, in: Proc. Analysis of Algorithms, 2005.
[14] L. Kharevych, P. Schroeder, B. Springborn, Discrete conformal mappings via circle patterns, ACM Trans. Graph. 25 (2) (2006) 412–438.
[15] R. Kleinberg, Geographic routing using hyperbolic space, in: Proc. INFOCOM, 2007.
[16] P. Koebe, Kontaktprobleme der konformen Abbildung, Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Kl. 88 (1936) 141–164.
[17] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: Proc. Canadian Conf. on Comp. Geom., 1999, pp. 51–54.
[18] A. Moitra, T. Leighton, Some results on greedy embeddings in metric spaces, in: Proc. FOCS'08, 2008, pp. 337–346.
[19] C. Papadimitriou, D. Ratajczak, On a conjecture related to geometric routing, in: Proc. ALGOSENSORS, 2004, pp. 9–17.
[20] R. Sarkar, F. Luo, X. Yin, X.D. Gu, J. Gao, Greedy routing with guaranteed delivery using Ricci flows, in: Proc. Symp. Inf. Proc. in Sensor Networks, 2009.
[21] A. Sheffer, B. Lévy, M. Mogilnitsky, A. Bogomyakov, ABF++: fast and robust angle based flattening, ACM Trans. Graph. 24 (2) (2005) 311–330.
[22] K. Stephenson, Introduction to Circle Packing. The Theory of Discrete Analytic Functions, Cambridge University Press, 2005.
[23] W. Thurston, The finite Riemann mapping theorem, in: Proc. Int. Symp. on the Occasion of the Proof of the Bieberbach Conjecture, Purdue Univ., 1985.

Please cite this article in press as: M. Ben Chen, et al., Distributed computation of virtual coordinates for greedy routing in sensor networks, Discrete Applied Mathematics (2010), doi:10.1016/j.dam.2010.10.016