Compass Routing on Geometric Networks

Evangelos Kranakis, School of Computer Science,
Carleton University, Ottawa, Canada
Harvinder Singh, School of Information Technology and
Engineering, University of Ottawa, Ottawa, Canada K1N 6N5
Jorge Urrutia, School of Information Technology and
Engineering, University of Ottawa, Ottawa, Canada K1N 6N5

1 Extended Abstract

Suppose that a traveler arrives in the city of Toronto, and wants to walk to the famous CN Tower, one of the tallest free-standing structures in the world. Our visitor, lacking a map of Toronto, is standing at an intersection from which he can see the CN Tower, and several streets $S_1, \ldots, S_m$ from which he can choose to start his walk. A natural (and most likely safe) assumption is that our visitor should choose the street whose direction points closest to the CN Tower; see Figure 1.

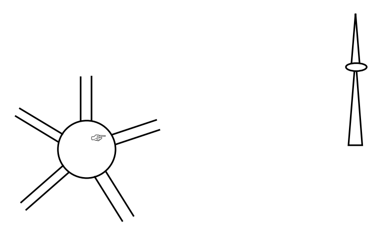

Figure 1: Finding our way to the CN Tower.

A close look at maps of numerous cities around the world shows us that using this method to explore a new and unknown city will, in general, yield walks that will be close enough to the optimal ones to travel from one location to another.

In mathematical terms, we can model the maps of many cities by geometric graphs in which street intersections are represented by the vertices of our graphs, and streets by straight line segments. Compass routing on geometric networks, in its most elemental form yields the following algorithm:

Compass Routing *Suppose that we want to travel from an initial vertex s to a destination vertex t, and that all the information available to us at any point in time is: the coordinates of our destination, our current position, and the directions of the edges incident with the vertex at which we are located. Starting at s, we will in a recursive way choose to traverse the edge of the geometric graph incident to our current position and with the closest slope to that of the line segment connecting the vertex of our current position to t. Ties are broken randomly.*

Using this criterion in the graph shown in Figure 2, if we want to travel from s to t, compass routing will produce the path s, a, b, c, t.

In this paper we study *local routing*

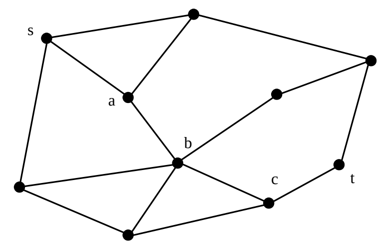

Figure 2: Traveling from s to t using compass routing.

algorithms on geometric networks. A routing algorithm is called a *local routing algorithm* if it satisfies the following conditions:

1. At each point in time, we know the coordinates of our starting position, as well as those of our destination. In addition, we have at our disposal a finite amount of storage where we can keep a constant number of identifiers of vertices of our network. Notice that this implies that at no point in time do we have full knowledge of the topology of the entire network.

2. Upon arrival at a vertex v (starting at s), we can use local information stored in v regarding its neighbours, and the edges connecting v to them. Using this information plus that stored in our local memory, we choose an edge incident to v, and traverse it until we reach its second end vertex, unless $v = t$, in which case we stop.

3. We are not allowed to change the local information stored at v. Notice that in particular, once we have left a vertex, if we return to it, we will not know that we have already visited it.

The motivation for the last condition can arise naturally when information is sent between different nodes of a network. For example if a server is connected to the web, we would like to avoid keeping track of the messages that have passed through our server, for this would easily use an enormous amount of memory that could quickly overload the storage available at those sites.

An approach to obtain local routing algorithms has been studied for distributed networks for which *compact routing* algorithms such as interval routing [11], boolean routing [9] etc. have been developed. Such schemes, however, can be worst-case storage-intensive in the sense that large amounts of information may be required be stored per node in order to achieve all-pair shortest path routing; see [3, 8]. Another drawback of the previous approach, perhaps more serious from our point of view, is that the topology of the networks for which these algorithms have been developed is assumed to be of a specific type, e.g. Cayley graphs. Our goal in this paper is that of developing routing algorithms that can be applied to existing communication networks whose only restriction is that they are planar networks. It is interesting to note that some of the best network topology maps used by internet service providers and internet backbone networks, such as TEN-34, EuropaNET, Eunet, Qwest Nationwide Network, and others, can be modeled as planar or almost planar graphs; see [1].

Finally, we mention that routing algorithms of a similar type to those studied in this paper have been studied within the framework of wireless communication networks, i.e. networks in which processors represent devices similar to radio stations, two of which can communicate if they are sufficiently close to each other; see [2, 7, 12]. In these problems, as in ours, the goal is not necessarily that of finding the shortest path connecting two vertices of the network, but to ensure that the information being transmitted does reach its destination.

It is not true however, that compass routing will always find a path from any starting point in a geometric graph to any other, not even in cases when our geometric graphs are 3-connected, and the internal as well as the external faces are bounded by convex polygons. The geometric graph shown in Figure 3 has these properties, yet when we try to go from $s = u_0$ to t using compass routing, we enter the cycle with vertex set $\{u_0, w_i; i = 0, \ldots, 5\}$, and continue to travel around it in an endless loop. Our graph consists of two concentric regular hexagons, one of which is rotated slightly with respect to the other. The line segment $t - u_i$ is orthogonal to the edge joining u_i to w_i and w_i lies on $t - u_i$. Also, u_{i+1} and v_i lie on the same side of the line through w_i and u_{i+1}, $i = 0, \ldots, 5$. It is now easy to see that under these conditions, if we are at point u_i (resp. w_i), compass routing will choose the edge connecting u_i to w_i (resp. w_i to u_{i+1}, addition taken mod 6).

We say that a geometric graph G supports compass routing if for every pair of its vertices s and t, compass routing (starting at s) produces a path from s to t. In a similar way we say that a planar graph G supports compass routing if there is an embedding of it (which produces a geometric graph)

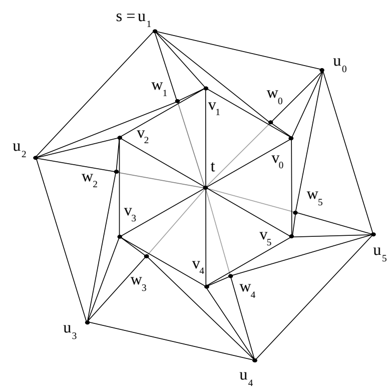

Figure 3: Compass routing will not reach t from u_i, $i = 0, \ldots, 5$.

which supports compass routing.

We prove that Delaunay triangulations of point sets on the plane support compass routing. We also obtain a geometric *local routing algorithm* that always finds a path between any two vertices of a connected geometric graph. We study the problem of determining which planar graphs have a geometric embedding that supports *shortest path compass routing*, i.e. embeddings of graphs for which compass routing actually produces the shortest path between any pair of its vertices. We prove that trees always have embeddings that support shortest path compass routing, and that not all outerplanar graphs, and hence not all planar graphs, have these types of embeddings.

References

[1] An Atlas of Cyberspaces, *http://www.geog.ucl.ac.uk/casa /martin/atlas/isp_maps.html.*

[2] Basagni, S., I. Chlamatac, V.R. Syrotiuk, and B.A. Woodward, "A distance routing effect algorithm for mobility", Proc. MOBICOM, 1998, 76-84.

[3] P. Fraigniau and C, Gavoille, "Universal routing schemes", *Journal of Distributed Computing*, **10** (1997), pp. 65-78.

[4] G. N. Frederickson and R. Janardan, "Designing networks with compact routing tables", *Algorithmica*, **3** (1988), pp. 171-190.

[5] G. N. Frederickson and R. Janardan, "Efficient message routing in planar networks", *SIAM Journal of Computing*, **18** (1989), 843-857.

[6] C, Gavoille and S. Perennes, "Lower bounds for shortest path interval routing", Proceedings of SIROCCO96, pp. 88-103, N. Santoro and P. Spirakis, eds, Carleton University Press, 1997.

[7] Ko, Y.B., and N.H. Vaidya, "Location-aided routing in mobile ad hoc networks", Proc. MOBICOM, 1998, 66-75.

[8] E. Kranakis and D. Krizanc, Proceedings of STACS96, pp. 529-540, C. Puech and R. Reischuk, eds, SVLNCS vol. 1046, 1996.

[9] E. Kranakis and D. Krizanc,"Boolean routing on Cayley networks", Proceedings of SIROCCO96, 119-124, N. Santoro and P. Spirakis, eds, Carleton University Press, 1997.

[10] Preparata, F., and I.M. Shamos, *Computational Geometry, an Introduction*. Springer Verlag, 1985.

[11] Santoro, N., and R. Khatib, "Labeling and implicit routing in networks" *The Computer Journal*, **28**, 1 (1985), 5-8.

[12] Stojmenovic, I., and X. Liu, "Geographic distance routing in ad hoc wireless networks", Preprint, SITE, University of Ottawa, 1999.