ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing Group

HS 2009

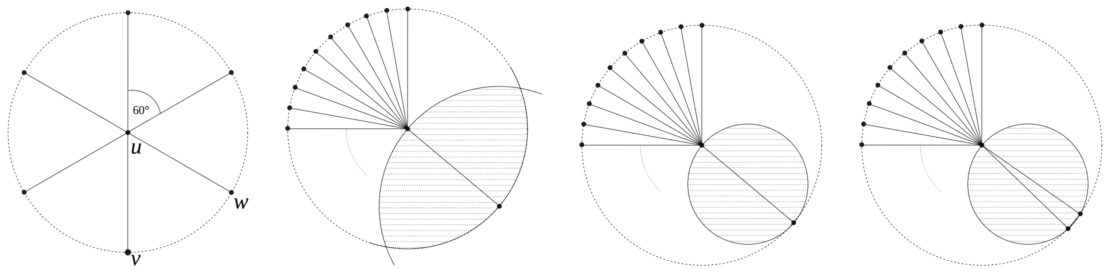
Prof. Dr. Roger Wattenhofer, Philipp Sommer, Johannes Schneider

Ad Hoc And Sensor Networks
Sample Solution to Exercise 3

Assigned: October 5, 2009
Due: October 12, 2009

1 Degree of Euclidean Graphs

From the four considered algorithms the Minimum Spanning Tree (MST) is the only one resulting in a bounded-degree topology. Figure 1 depicts worst-case instances for the different graphs. Figure 1(a) gives thereby an idea why the degree of the MST is bounded to six. If the angle between two adjacent edges (u, v) and (u, w) is less than 60 degrees it would be cheaper to abandon one of these edges and add (v, w) to the final topology. By the definition of the Relative Neighborhood Graph (RNG), an edge is discarded if a node is situated inside the lune of this edge (see Figure 1(b)). However, the boundary is not included. If we therefore arrange n nodes on a circle around another node, we end up with a maximum degree of n. The same worst-case example also exists for the Gabriel Graph (GG), and the Delaunay Triangulation (DT) (Figure 1(c) and (d)). With these topologies the critical areas depicted in Figure 1 do not contain any other nodes on the circle around the center node. Thus, we also obtain a graph with unbounded degree in the worst case.



(a) MST (b) RNG (c) GG (d) DT

Figure 1: Bad network instances for the four Euclidean graphs. One can see that only the MST has bounded degree.

2 Face Routing continued

a) Consider Figure 2, where node A and B are connected by an edge of length 1, C and D are in the circle spanned by $\overline{AB}$, and $|BC| > d$ and $|DA| > d$. By the QUDG connectivity model, the edges $\overline{BC}$ and $\overline{DA}$ need not be present.

   When the Gabriel Graph is applied directly to the graph, node A will include the edge $\overline{AC}$, but not $\overline{AB}$, as C is in the circle spanned by $\overline{AB}$. Similarly, B includes the edge $\overline{BD}$, but not $\overline{BA}$, as D is in the circle spanned by $\overline{BA}$. The resulting Gabriel Graph consists only of the two edges $\overline{AC}$ and $\overline{BD}$, disconnecting the originally connected graph.

b) The definition of an SDG says that the end-points of any connection (u, v) are connected to any node contained in the disk with diameter (u, v). In fact, that is all information the Gabriel Graph needs to know to planarize a given graph. Therefore, we can directly apply

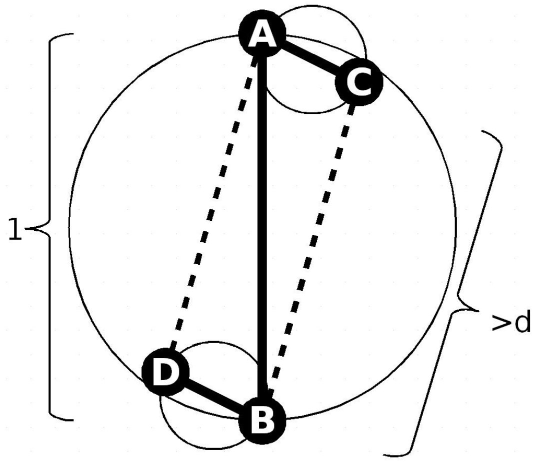


Figure 2: Gabriel Graph on QUDGs may disconnect the graph.

the Gabriel Graph on any SDG to obtain a planar subgraph, on which we can run our face routing algorithms.

3 Gabriel Graph Spanner Property

It can be shown that the Gabriel Graph (GG) is no Euclidean spanner but has a spanner ratio in $\Omega(\sqrt{n})$. In Figure 2 a bad network instance is depicted. In order to clarify the example we assume that the construction of the GG is done in steps. We first consider the edge (u, v). We see that this edge is replaced by a detour via an additional node (indicated by the two arrows labeled with 1). However these edges are also replaced in a second step. If we develop this recursive concept, one can see that we obtain a fractal-like construction. We now use the Pythagorean theorem to compute the total length of the path from u to v along the GG.

We make the following observations:

- With the given construction scheme, the GG degenerates to a chain. Thus, the total number of hops on the GG from u to v is $n - 1$
- In each construction step the number of edges in the graph doubles as each existing edge is replaced by two new ones. As a consequence the construction algorithm terminates in $O(\log n)$ rounds.

By applying the Pythagorean theorem we can compute the Euclidean length of the path from u to v during each round of the construction as:

$$l_o = \overline{uv} = 1, l_1 = 2 * 1/\sqrt{2}, l_2 = 4 * 1/\sqrt{2}^2, \ldots, l_i = 2^i * 1/\sqrt{2}^i$$

We apply $i = \log n$ and get a total Euclidean length of

$$2^{\log n} * 1/\sqrt{2}^{\log n} = n * 1/\sqrt{n} = \sqrt{n}$$

By definition the shortest path between any two points in a spanner graph must not be more than a constant factor longer than the shortest path between these two nodes in the original graph. As we have shown the GG does not satisfy this property as it may produce a path which is $O(\sqrt{n})$ times longer than the shortest path. Thus, the GG is *no* Euclidean spanner.

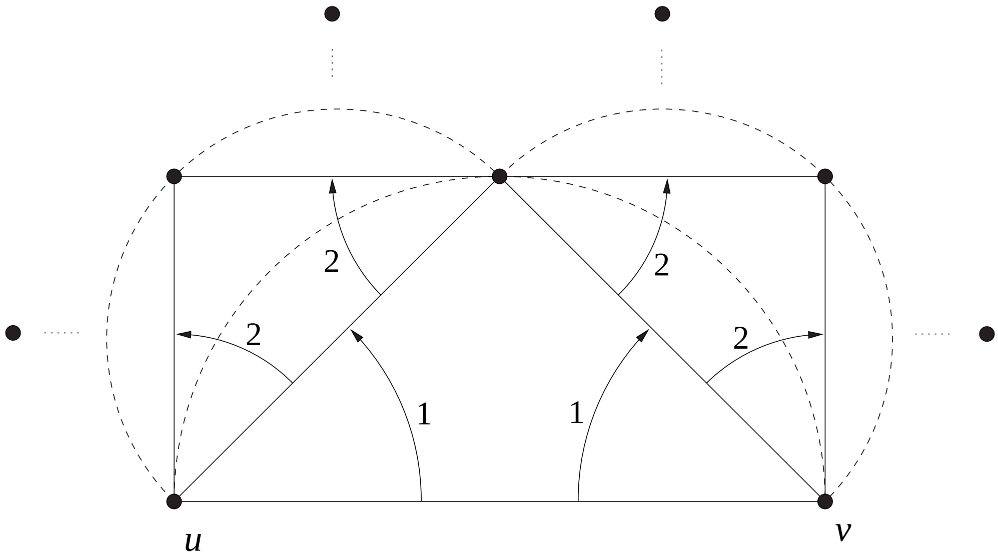


Figure 3: Fractal construction of a worst-case instance for the Gabriel Graph.

4 Topology Control

a) The benefit of topology control is more pronounced in dense networks as opposed to sparse topologies: Redundancy and high congestion normally come along with high density. From a management point of view it is better to restrict the number of neighbors to as few as possible. This not only facilitates routing decisions but also helps bounding the size of the required neighbor table. As it was shown in the lecture topology control also allows nodes to reduce their sending power. On the one hand, this leads to energy conservation at the individual nodes. On the other hand, by decreasing transmission power interference is diminished yielding higher throughput and less retransmissions due to collisions. However, topology control does also make sense in sparse networks to get rid of costly links that are not vital for network connectivity.

   The overhead incurred by a topology algorithm should be kept in mind. In order to maintain the desired properties of the network the algorithm has to exchange information with all of its potential neighbors. This includes periodic quality measurements of individual links and the actual distributed computation of the algorithm itself to handle dynamic environments. This overhead may become significant if not much data is sent over the network. As a consequence, the algorithm wastes a lot of energy that could be used by the application itself if we completely discarded topology control. Another issue that leads to a bad overall network performance is the establishment of virtual bottlenecks. Topology control may drop links that are not required to satisfy the desired properties. However, some of the abandoned links might allow extra routes between almost independent components (clusters) in the network. Without them network traffic between the components is burdened on the remaining links connecting the two parts of the network.

b) We show that the XTC algorithm results in a bounded degree topology for UDGs by proving that no two adjacent edges in the network enclose an angle less than $\pi/3$. From this it follows that XTC yields degree at most 6. Assume for contradiction that the two edges (u, v) and (u, w) enclose an angle $\alpha < \pi/3$ at node u. Furthermore let v be u's neighbor that was included into u's neighbor list before w, that is $v \prec_u w$ or $dist(u, v) < dist(u, w)$. Since the distance between u and v is less than the distance between u and w and $\alpha < \pi/3$, it follows that $dist(v, w) < dist(u, w)$. Since we consider a UDG, also the edge (v, w) is in the graph, as $dist(v, w) < dist(u, w) < 1$. Consequently $v \prec_w u$, implies that u is not included in w's neighbor list, which is however a contradiction to the assumption that the edge (u, v) is in the resulting topology.

   If we abandon the constraint that the given network is a Unit Disk Graph, the XTC algorithm is no longer able to guarantee bounded degree. It can be seen from Figure 3 that the angle

α between two links of node v can become arbitrary small in the presence of obstacles.

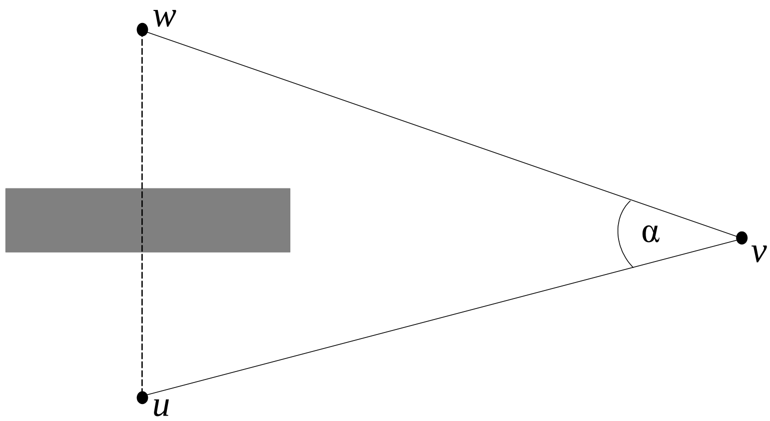


Figure 4: XTC cannot guarantee a minimum angle α, and thus also no bounded degree, if obstacles are considered.