CS250B Homework 4

Due Thursday, April 9, 2009, at 2:30 pm

1 Rightmost shortest-path tree

Give a linear-time algorithm for the following problem:

- input: a directed planar embedded graph with edge-lengths, a vertex r on the boundary of the infinite face, and a table giving distances from r to all vertices.
- output: a rightmost shortest-path tree rooted at r.

2 Need for evert

Give an explanation using diagrams showing why the dynamic tree used in implementing the MSSP algorithm needs to support evert.

3 Fast recursive decomposition

Give an algorithm for the following problem:

- input: an undirected planar embedded graph G_0, a rooted spanning tree T.
- output: a recursive decomposition.

The separators used for the recursive decomposition should be elementary cycles with respect to T. The separator C for graph G should be good in that the number of edges strictly enclosed by C should be at most a constant fraction of the number of edges of G, and the number of edges not enclosed by C should be at most a constant fraction of the number of edges of G.

The leaves of the recursive decomposition should correspond to "small" graphs in some sense.

It is straightforward to achieve this in $O(m \log m)$ where m is the number of edges in G_0. Partial credit will be awarded for a clear, simple argument that $O(m \log m)$ is achievable. Full credit will be awarded for an algorithm whose running time is $o(m \log m)$, i.e. asymptotically faster.

4 Minimum st-cut in an undirected planar embedded graph

For an undirected graph G and two vertices s and t, a cut $\delta(S)$ is an *st-cut* if $s \in S$ and $t \notin S$. Suppose the edges are assigned nonnegative numbers (*capacities*). A *minimum-capacity st-cut* (or just *minimum st-cut*) is an *st*-cut of minimum total capacity.

Your goal is an $O(m \log m)$ algorithm for the following problem:

- input: undirected planar embedded graph G with nonnegative edge-capacities, vertices s and t
- output: a minimum-capacity set of edges separating s and t

To start you off, I'll describe an algorithm for a simpler problem, one where we require that there is a zero-capacity edge e between s and t. Let G^* be the dual graph. Let x and y be the endpoints of the edge e in G^*. Find a shortest x-to-y path P^* in the graph obtained from G^* by deleting e, where we interpret capacities as lengths. The path P^*, together with the edge e, is a simple cycle in G^*. The construction ensures that it is the minimum-length cycle that encloses the face s and not the face t (or vice versa). It therefore corresponds to a cut in G that separates s and t, in particular the minimum-capacity such cut.

Remarks:

- The simpler case can be solved in linear time (though using techniques we haven't yet covered).
- For the original problem, no algorithm is known that has a running time asymptotically less than $O(m \log m)$.

5 Maximum st-flow in undirected planar graphs where s and t are adjacent

For a graph G with vertices s and t, an *st-flow* is a vector $\boldsymbol{f}$ in arc space such that, for every vertex v other than s and t, $\boldsymbol{\delta}(v) \cdot \boldsymbol{f} = 0$. The *value* of the flow is $\boldsymbol{\delta}(s) \cdot \boldsymbol{f}$.

A *capacity function* is an assignment $c(\cdot)$ of numbers to darts. We say that a vector $\boldsymbol{\eta}$ is *feasible* with respect to $c(\cdot)$ if, for every dart d, $\boldsymbol{\eta}[d] \leq c(d)$.

Given a graph G with a capacity function $c(\cdot)$, and given vertices s and t, a *maximum st-flow* is a feasible *st*-flow (i.e. feasible with respect to $c(\cdot)$) whose value is maximum.

For an undirected graph, it is customary to require that the capacity function be symmetric, i.e. $c(d) = c(\text{rev}\,(()d))$ for every dart d, and we will make that assumption here.

It is easy to see that the value of any *st*-flow is at most the capacity of any *st*-cut. Therefore, for any *st*-flow f and *st*-cut $\delta(S)$ such that the value of $\boldsymbol{f}$ equals the capacity of $\delta(S)$, it must be that $\boldsymbol{f}$ is a *maximum st*-flow and $\delta(S)$ is a minimum *st*-cut.

To make this problem simpler (and independent on the solution to the previous problem), we will restrict our attention to undirected planar embedded graphs G where there is a zero-capacity edge between s and t. Use the idea of the min st-cut algorithm described in the previous problem to give an algorithm for max st-flow in this case.

Now for the hint:

Recall that a *circulation* in G is a vector in the cycle space of G, i.e. a vector $\boldsymbol{\eta}$ in the arc space of G such that $\boldsymbol{\delta}(v) \cdot \boldsymbol{\eta} = 0$ for every vertex v.

Recall that one basis for the cycle space of G is the face basis; for each face f except the infinite face, the basis includes $\boldsymbol{\delta}(f)$.

Lemma: Let G be an undirected planar graph with a symmetric, nonnegative edge-capacity function $c(\cdot)$. For each face f, let $d(f)$ be the f_∞-to-f distance in G^*, where the length of an edge is its capacity in the primal. Let $\boldsymbol{\eta}$ be the vector in the cycle space of G that is the linear combination

$$\sum_{f \neq f_\infty} d(f)\boldsymbol{\delta}(f)$$

of face-basis vectors.
Then $\boldsymbol{\eta}$ is feasible with respect to $c(\cdot)$.