Topics in Algorithms and Data Structures

Notes and exercises for May 20

Christian Wulff-Nilsen *

1 Max flow in planar graphs

We consider the max flow problem for directed planar graphs. Even the fastest max flow algorithms for general graphs require roughly quadratic time when applied to planar graphs. We will look at Hassin's planarity-exploiting max flow algorithm [1] which significantly improves this bound in the special case where the planar graph is embedded in the plane (a so called *plane graph*) with the source and sink contained in the same face.

2 Duality

In order to discuss Hassin's algorithm, we need the concept of duality for weighted directed plane graphs. Let $G = (V, E, c : E \to \mathbb{R})$ be an n-vertex directed plane graph with a (non-negative) edge capacity function c. We will make the assumption that two oppositely directed edges (u, v) and (v, u) are embedded so that they coincide in the plane (i.e., there is a single curve defining the embedding of both edges). The dual graph G^* has a vertex for every face of G. We typically identify vertices of G^* with these faces. For every directed edge $e \in G$, G^* has an edge e^*, defined as follows. Let f_r resp. f_l be the face of G to the right resp. left of e when viewed in the direction of e. Then the dual edge e^* is directed from f_r to f_l. A weight $w(e^*)$ is assigned to e^* which is equal to the capacity $c(e)$ of primal edge e. In general, G^* is a multigraph, i.e., it may contain self-loops and multiple edges between the same ordered pair of vertices. It can be shown that G^* has $O(n)$ vertices and edges.

2.1 Duality of cuts and cycles

An important property used in Hassin's algorithm is the duality of cuts in the primal and separating cycles in the dual. First, we need a new definition. Let H be a directed plane graph with non-negative edge weights and let f_1 and f_2 be two distinct faces in H. Then a simple cycle C in H is called *f_1f_2-separating* if f_1 and f_2 are on different sides of C (when viewing C as a curve in the plane). We also say that C *separates* f_1 and f_2. A *min f_1f_2-separating cycle* is a shortest f_1f_2-separating cycle with respect to the edge weight function.

With G defined as above, assume it is connected. Then it can be shown that $G = (G^*)^*$ so a vertex u in G corresponds to a face u^* in G^*. For distinct vertices s and t in G, it is easy

*Department of Computer Science, University of Copenhagen, `koolooz@di.ku.dk`, `http://www.diku.dk/~koolooz/`.

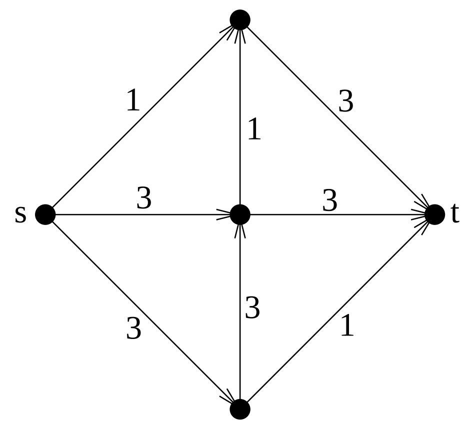

Figure 1: A plane directed graph with edge capacities and a source s and a sink t belonging to the external face.

to prove that the edges of a min s^*t^*-separating cycle C in G^* defines a min st-cut in G if C wraps around s^* in a counter-clockwise order or wraps around t^* in a clockwise order; here we represent the min st-cut by the set of edges crossing it from the s-side to the t-side of the cut. Also, the weight of such a cycle C equals the capacity of a min st-cut in G.

3 Hassin's algorithm

Hassin's algorithm [1] solves the following problem: for a source s and a sink t belonging to the same face f of an n-vertex plane directed graph $G = (V, E, c : E \to \mathbb{R})$ with edge-capacity function c, find a max st-flow in G[1]. The algorithm exploits a surprising connection between max st-flow in G and shortest paths in G^*. We will give a slightly different description of the algorithm than that in [1].

First, a directed edge e_∞ of infinite capacity is added to G from t to s inside f. This splits f into two faces f_l and f_r, where f_l resp. f_r is to the left resp. right of e_∞. Dual graph G^* is updated accordingly. Then a single source shortest path tree T^* in G^* with source f_r^* is found. For each dual vertex u^*, assign the potential $\Phi(u^*) = d_{G^*}(f_r^*, u^*)$ where $d_{G^*}(x, y)$ denotes the shortest path distance between two vertices x and y in G^*. Each edge e in G is then assigned the flow value $f(e) = \Phi(f_2^*) - \Phi(f_1^*)$, where $e^* = (f_1^*, f_2^*)$ is the directed dual edge corresponding to e. It can be shown (see Exercise 2) that f is a max st-flow in G. Using Dijkstra's algorithm to compute a single-source shortest path tree in G^*, Hassin's algorithm runs in $O(n \log n)$ time. This bound can be improved to linear by instead using the linear time single-source shortest path tree algorithm in [2].

Exercises

Exercise 1: dual graph and Hassin's algorithm

1. Draw the dual (with edge orientations and edge weights) of the graph in Figure 1 augmented with edge e_∞.

[1] Hassin calls a plane graph where s and t are on the same face an (s,t)-*planar network*.

2. Find a max st-flow in the graph in Figure 1 using Hassin's algorithm.

Exercise 2: correctness of Hassin's algorithm

Now, consider applying Hassin's algorithm to a general plane directed graph $G = (V, E, c : E \to \mathbb{R})$ with a source s and a sink t belonging to the same face f. We will prove that a max st-flow in G is found.

In this exercise, it is useful to think of every edge in G (and hence also in G^*) as bidirected (for instance, for each edge in Figure 1, there is implicitly a reverse edge of capacity 0). Also, sending $f(v, w)$ units of flow along primal edge (v, w) corresponds to sending $-f(v, w)$ units of flow along (w, v), i.e., $f(w, v) = -f(v, w)$.

1. Let e be any edge of G and let $e^* = (f_1^*, f_2^*)$ be the corresponding dual edge in G^*. Show that $\Phi(f_2^*) - \Phi(f_1^*) \leq c(e)$. Conclude that capacity constraints are satisfied.

2. Show that for any cycle C in G^* with edge set $E^*(C)$, $\sum_{(f_1^*, f_2^*) \in E^*(C)} (\Phi(f_2^*) - \Phi(f_1^*)) = 0$. Conclude that there is flow conservation at any vertex $u \in V \setminus \{s, t\}$ after removing e_∞.

3. Let f_l and f_r be defined as in Section 3. Show that the edges of a shortest path P from f_r^* to f_l^* in G^* define a min st-cut in G. In your answer, you may assume the correctness of the results in Section 2.1.

   Hint: removing e_∞ from G can be regarded as merging f_r and f_l to the original face f of G containing s and t. What happens with the endpoints of P in the process? Feel free to ignore details on cycle orientations in your proof.

4. Show that all edges of this min st-cut are saturated with respect to the flow assigned by Hassin's algorithm. That is, show that $f(e) = c(e)$ for each primal edge e corresponding to a dual edge e^* of P.

5. Conclude that Hassin's algorithm finds a max st-flow in G.

References

[1] R. Hassin. Maximum flows in (s, t) planar networks. Information Processing Letters, Vol. 13, No. 3, pp. 107, 1981.

[2] M. R. Henzinger, P. N. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. J. Comput. Syst. Sci., 55(1):3–23, 1997.