CS 598 JGE Project Proposal
Kyle Fox

Maximum flows and minimum cuts are some of the most heavily studied areas in theoretical computer science. Since the publication of Ford and Fulkerson's paper [7] which introduced the max-flow min-cut theorem, much work has been done to provide faster algorithms for computing maximum flows on graphs. In particular, work has been done to find more efficient algorithms for maximum flows and minimum cuts in planar undirected [8, 10, 12] and directed [1, 2, 6] graphs. Until very recently, though, nothing was known about computing maximum flows and minimum cuts in graphs embedded on more general surfaces that was not simply based on graph sparsity. This changed when Chambers *et al.* showed an $O(n \log n)$ algorithm for computing a minimum cost cut in an undirected planar graph embedded on a surface of fixed genus [4] as well as efficient algorithms for computing maximum flow in a directed graph embedded on a surface of fixed genus [3].

Despite this recent success, it appears that there is still work to be done before we can match the running time of Borradaile and Klein for planar graphs [2]. For a graph of genus g, the algorithms of [3] run in $O(g^7 n \log^2 n \log^2 C)$ if we are given integer capacities summing to C and $O(g^{O(g)} n^{3/2})$ if a combinatorial algorithm is required. Chambers *et al.* conjecture in the same paper that this can be improved to $O(g^k n \log n)$ for some small constant k, but it is still unclear how we might accomplish this.

We formally define the problem as follows using notation given in [6]. We are given a directed graph $G = (V, E)$ embeddable on a surface Σ with designated source and sink vertices $s, t \in V$. Assume that for every edge $(u \to v) \in E$, there also exists the reversal edge $rev(u \to v) = (v \to u)$. An (s, t)-flow is a function $\phi : E \to \mathbb{R}$ such that $\phi(e) = -\phi(rev(e))$ for every edge e and $\sum_w \phi(v \to w) = 0$ for every vertex v except s and t. The value of the flow is $\sum_w \phi(s \to w)$. If we are also given non-negative edge capacities $c : E \to \mathbb{R}$, then ϕ is considered feasible if $\phi(e) \leq c(e)$ for every edge e. We wish to find a feasible flow of maximum value.

Open Problem 1. *Does there exist a combinatorial $O(g^k n \log n)$ maximum flow algorithm for graphs embeddable on a surface of genus g where k is some small constant? Baring that, does there exist an $O(f(g) n \log n \operatorname{polylog} C)$ algorithm for embeddable graphs with integer edge capacities where f is a function of the genus and C is the sum of the edge capacities?*

One seemingly promising line of attack was explored by Erickson in his restating of Borradaile and Klein's solution to the planar maximum flow problem in an input graph G as a parametric shortest path problem for the oriented dual graph G^* [6]. Changes to the shortest path tree in the dual graph can be effected in $O(\log n)$ time using a dynamic tree data structure such as a self-adjusting top tree [13]. The tree only has to change $O(n)$ times, so the total running time of the algorithm is $O(n \log n)$. Unfortunately, the number of changes to the tree is no longer linear when the graph is embedded on a surface of higher genus, supporting the idea that other tactics will be necessary to solve the problem.

Another potential line of attack is to consider planar subgraphs of the input graph. In particular, suppose our input graph is embedabble on a surface of genus g and the edges are given integer capacities. It may be possible to find some planar subgraph $P \subseteq G$ such that $maxflow(P) \geq maxflow(G)/f(g)$ for some function f of the genus. By repeatedly finding augmenting flows through these planar subgraphs using an $O(n \log n)$ planar maximum flow algorithm, it should be possible to compute a maximum flow for G in $O(f(g)n \log n \log |F^*|)$ where $|F^*|$ is the value of the maximum flow in G. While this algorithm is not combinatorial due to the dependence on integer edge weights, it would remove a $\log n$ factor from the running time of the current best known algorithm for surfaces of bounded genus. Unfortunately, we do not know how to find these special planar subgraphs or if they even exist.

It is also conceivable that we could use graph separators to solve the problem since any surface graph of bounded genus has a small separator [5]. Separators are used for finding planar shortest paths with non-negative edge lengths in linear time [11], so it seems likely that it could be used for flows in higher genus graphs as well. It may also be possible to decompose the input graph into a number of planar subgraphs so we may perform optimizations of flow traveling over and between these subgraphs. Perhaps the push/relabel strategies of Goldberg and Tarjan could be used while computing flows between the subgraphs [9].

References

[1] Glencora Borradaile. *Exploiting Planarity for Network Flow and Connectivity Problems*. PhD thesis, Brown University, May 2008.

[2] Glencora Borradaile and Philip Klein. An $O(n \log n)$ algorithm for maximum *st*-flow in a directed planar graph. *J. ACM*, 56(2), 2009.

[3] Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Homology flows, cohomology cuts. In *Proc. 42nd Ann. ACM Symp. Theory Comput.*, pages 273–282, 2009. Full version available at http://www.cs.uiuc.edu/~jeffe/pubs/surflow.html.

[4] Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. In *Proc. 25th Ann. ACM Symp. Comput. Geom.*, pages 377–385, 2009.

[5] David Eppstein. Dynamic generators of topologically embedded graphs. In *Proc. 14th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 599–608, 2003.

[6] Jeff Erickson. Maximum flows and parametric shortest paths in planar graphs. In *Proc. 21st Ann. ACM-SIAM Symp. Discrete Algorithms*, to appear, 2010.

[7] Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian J. Math.*, 8(399–404), 1956. First published as Research Memorandum RM-1400, The RAND Corporation, Santa Monica, California, November 19, 1954.

[8] Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM J. Comput.*, 16(6):1004–1004, 1987.

[9] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.*, 35(4):921–940, 1988.

[10] Refael Hassin and Donald B. Johnson. An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks. *SIAM J. Comput.*, 14(3):612–624, 1985.

[11] Monika R. Henzinger, Philip Klein, Satish Rao, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55(1):3–23, 1997.

[12] John Reif. Minimum *s*-*t* cut of a planar undirected network in $O(n \log^2 n)$ time. *SIAM J. Comput.*, 12:71–81, 1983.

[13] Robert E. Tarjan and Renato F. Werneck. Self-adjusting top trees. In *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms*, pages 813–822, 2005.