Title:	Maximum st-Flow in Planar Graphs
Name:	Glencora Borradaile[1]
Affil./Addr.	School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA
Keywords:	Planar graphs; Maximum Flow
SumOriWork:	2010; Erickson
	2009; Borradaile, Klein
	2006; Borradaile, Klein

Maximum st-Flow in Planar Graphs

Glencora Borradaile[1]

School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA

Years aud Authors of Summarized Original Work

2010; Erickson
2009; Borradaile, Klein
2006; Borradaile, Klein

Keywords

Planar graphs; Maximum Flow

Problem Definition

Given a directed, planar graph $G = (V, E)$ with arc capacities $c : E \to \Re^+$, a source vertex s and a sink vertex t, the goal is to find a flow assigment f_e for each arc $e \in E$ such that:

$$
\begin{aligned}
\max \quad & \sum_{su:su\in E} f_{su} \\
s.t. \quad & \sum_{uv:uv\in E} f_{uv} - \sum_{vw:vw\in E} f_{vw} = 0 \qquad \forall v \in V \setminus \{s,t\} && (1)\\
& 0 \le f_e \le c_e \qquad \forall e \in E && (2)
\end{aligned}
$$

Key Results

In the paper proposing the maximum flow problem in general graphs, Ford and Fulkerson [5] gave a generic method for computing a maximum flow: the augmenting-path algorithm. The algorithm is iterative: find a path P from the source to the sink such that capacity contraint (2) is loose for each arc on P (*residual*); increase the flow on each arc in P by a constant chosen so that at least one of the capacity constraints become tight; update the capacities of each arc, making note that the reverse of these arcs now have *residual capacity*; repeat until there is no path from the source to the sink along which the flow can be augmented. By augmenting the flow along a path, the balance constraints (1) are always satisfied.

st-planar graphs

Ford and Fulkerson further showed that, in the case of planar graphs when the source and the sink are on a common face (st-planar graphs), by selecting the augmenting paths to be as far to the left as possible in each iteration (viewing s on the bottom and t on the top), each arc is saturated at most once, resulting in at most $|E|$ iterations [5]. In 1979, Itai and Shiloach showed that each iteration of this algorithm could be implemented in $O(\log n)$ time using a priority queue and gave a simple example showing that any implementation of this algorithm is capable of sorting n numbers [11]. In 1991, Hassin demonstrated that such a maximum st-flow could be derived from shortest-path distances in the planar dual G^* of G where capacities in G are interpreted as lengths in G^* [7]. Faster algorithms for computing shortest paths in planar graphs culminated in a linear-time algorithm for this case of maximum st-flow in planar graphs with s and t on a common face [9].

Undirected planar graphs

For undirected planar graphs, Reif gave an algorithm for computing the maximum st-flow where s and t need not be on a common face, by way of several shortest path computations in the dual [19]. The algorithm finds a shortest path P in G^* from a vertex adjacent to the face corresponding to s to a vertex adjacent to the face corresponding to t. Reif proves that C only crosses P once; by finding the minimum separating cycle C_v through each vertex v of P, we will surely find C: C is the minimum of the cycles C_v. These cycles can be found in time $\log n$ times the time for one shortest path computation via divide and conquer over the length of P. Hassin and Johnson show that the corresponding maximum flow can be computed within this framework by computing shortest path distances between the nested cycles C_v [8]. The shortest path algorithms of Henzinger et al. [9] or Klein [15] can be used to re-implement these algorithms in $O(n \log n)$ time. Italiano et al.[12] further improved this running time to $O(n \log logn)$ by using an r-division to break the graph into sufficiently small pieces through which shortest paths can be efficiently computed.

If the capacities are all unit, the maximum st-flow can be computed in linear time [1].

Directed planar graphs

Maximum st-flow in directed graphs is more general since the problem of maximum st-flow in an undirected graph can be converted to a directed problem by introducing two oppositely oriented arcs of equal capacity for each edge. Johnson and Venkatesan gave a divide-and-conquer algorithm that finds a flow of input value v in $O(n^{1.5} \log n)$ time [13]. The algorithm divides the graph using balanced separators, finding a flow in each side of value v. However, the flow on the $O(\sqrt{n})$-boundary edges of each subproblem might not be feasible. Each boundary edge is made feasible via an st-planar flow computation. Miller and Naor showed that finding a directed st-flow of value v could be reduced to computing shortest-path distances in a graph with positive and negatives lengths [17].

Here, v units of flow are routed (perhaps violating the capacity constraints) along any s-to-t path P. For those arcs whose capacity are violated, we must route the excess flow through the rest of the graph. This is a feasible circulation problem and can be solved using shortest-path distances in the dual graph, where lengths may be negative (representing the negative or violated capacities). Using an $O(n\, poly \log n)$-time algorithm for computing shortest paths in a planar graph with negative edge lengths [4; 16; 18] gives an $O(n\, poly \log n \log C)$-time algorithm where C is the sum of the capacities.

If the capacities are all unit, the maximum st-flow can be computed in linear time [21].

Leftmost-path algorithm

Borradaile and Klein gave an augmenting-path, $O(n \log n)$-time algorithm for the maximum st-flow problem in directed planar graphs. The algorithm is a generalization of the algorithm for the st-planar case, augmenting flow repeatedly along the leftmost path from s to t. However, with s and t not on a common face, what leftmost is not clear. With the graph embedded such that t is on the external face and the clockwise cycles saturated, a leftmost path is well-defined and can be found with a left-first, depth-first search into t. Clockwise cycles can be initially saturated with a circulation defined by potentials on the faces given by shortest-path distances in the dual graph [14] and clockwise cycles remain saturated under leftmost augmentations. Borradaile and Klein, and Erickson improved the analysis [3] showed that under these conditions an arc and its reverse can be saturated at most once, resulting in at most $2n$ augmentations. Augmentations can be performed in $O(\log n)$ time using a dynamic-tree data structure, resulting in an $O(n \log n)$ running time.

Applications

Maximum st-flow in directed planar graphs has applications to computer vision problems. Schmidt et al. [20] use it as a black box for image segmentation and Greig et al. [6] provide an example for smoothing noisy images.

Open Problems

Currently, maximum st-flow in undirected planar graphs can be computed more quickly than in directed. Can this gap be closed?

Experimental Results

Schmidt et al. [20] have implemented this algorithm and compared its performance on an image segmentation problem.

URLs to Code and Data Sets

Hoch and Wang have provided an open-source implementation of the algorithm [10]. Eisenstat has an implementation of the linear-time algorithm for unit-capacity graphs. [2].

Cross-References

Multiple source, multiple sink maximum flow in directed planar graphs
Maximum flow

Recommended Reading

1. Coupry L (1997) A simple linear algorithm for the edge-disjoint (s,t)-paths problem in undirected planar graphs. Information Processing Letters 64:83–86
2. Eisenstat D (2013) trickle: linear-time maximum flow in planar graphs with unit capacities (java). URL `http://www.davideisenstat.com/trickle/`
3. Erickson J (2010) Maximum flows and parametric shortest paths in planar graphs. In: 21st SODA, pp 794–804
4. Fakcharoenphol J, Rao S (2006) Planar graphs, negative weight edges, shortest paths, and near linear time. J Comput Syst Sci 72(5):868–889, DOI http://dx.doi.org/10.1016/j.jcss.2005.05.007
5. Ford C, Fulkerson D (1956) Maximal flow through a network. Canadian Journal of Mathematics 8:399–404
6. Greig D, Porteous B, Seheult A (1989) Exact maximum a posteriori estimation for binary images. J Roy Stat Soc B 51(2):271–279
7. Hassin R (1981) Maximum flow in (s,t) planar networks. Inform Process Lett 13(3):107
8. Hassin R, Johnson DB (1985) An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks. SIAM J Comput 14:612–624, DOI http://locus.siam.org/SICOMP/volume-14/art_0214045.html
9. Henzinger MR, Klein PN, Rao S, Subramanian S (1997) Faster shortest-path algorithms for planar graphs. J Comput Syst Sci 55(1):3–23, DOI 10.1145/195058.195092
10. Hoch J, Wang J (2012) Max flow in a directed planar graph. URL `https://github.com/jrshoch/msmsmaxflow`
11. Itai A, Shiloach Y (1979) Maximum flow in planar networks. SIAM J Comput 8:135–150
12. Italiano GF, Nussbaum Y, Sankowski P, Wulff-Nilsen C (2011) Improved algorithms for min cut and max flow in undirected planar graphs. In: 43rd STOC, pp 313–322
13. Johnson DB, Venkatesan SM (1983) Partition of planar flow networks. In: 24th FOCS, pp 259–264, DOI 10.1109/SFCS.1983.44
14. Khuller S, Naor J, Klein P (1993) The lattice structure of flow in planar graphs. SIAM J Discret Math 6(3):477–490, DOI 10.1137/0406038
15. Klein PN (2005) Multiple-source shortest paths in planar graphs. In: 16th SODA, pp 146–155, DOI 10.1145/1070454
16. Klein PN, Mozes S, Weimann O (2010) Shortest paths in directed planar graphs with negative lengths: A linear-space $O(n \log^2 n)$-time algorithm. TALG 6(2):1–18, DOI http://doi.acm.org/10.1145/1721837.1721846
17. Miller GL, Naor J (1995) Flow in planar graphs with multiple sources and sinks. SIAM J Comput 24(5):1002–1017, DOI 10.1137/S0097539789162997
18. Mozes S, Wulff-Nilsen C (2010) Shortest paths in planar graphs with real lengths in $O(n \log^2 n / \log \log n)$ time. In: 18th ESA, pp 206–217
19. Reif J (1983) Minimum s-t cut of a planar undirected network in $O(n \log^2 n)$ time. SIAM J Comput 12:71–81, DOI http://locus.siam.org/SICOMP/volume-12/art_0212005.html
20. Schmidt FR, Toeppe E, Cremers D (2009) Efficient planar graph cuts with applications in computer vision. In: CVPR, pp 351–356
21. Weihe K (1994) Edge-disjoint (s,t)-paths in undirected planar graphs in linear time. In: Proc. of the European Symp. on Algorithms, LNCS, vol 855, pp 130–137