Approximation Algorithms Workshop June 13-17, 2011, Princeton

Lecture New Approximation Schemes for Optimization Problems in Planar Graphs —

Philip N. Klein *Scribe: David Eisenstat*

1 Overview

We all know the world is flat, but it's not Euclidean. If you start with these points in the plane and find a Euclidean traveling salesman tour, you'll probably get wet. You really want to find tours in a planar graph. Research goal: exploit planarity to achieve faster algorithms and more accurate approximations.

For example,

faster algorithms shortest paths, maximum flow

more accurate algorithms traveling salesman, steiner tree, multiterminal cut

There is synergy between these goals.

Example of faster algorithm: multiple-source shortest paths. $O(n \log n)$ algorithm to compute all of the shortest path trees from vertices on the boundary. (We must use an implicit representation because there may be $\Omega(n)$ boundary vertices.)

Approximation schemes:

Lipton–Tarjan 1977 maximum independent set in $O(n \log n)$

Baker 1983 max independent set, partition into triangles, min vertex-cover, min dominating set. . . in $O(n)$

Grigni–Koutsoupias–Papadimitriou 1995 Traveling salesman in unweighted graphs $n^{O(1/\epsilon)}$

Arora–Grigni–Karger–Klein–Woloszyn 1998] Traveling salesman in graphs with weights

Definition 1. *An approximation scheme is* efficient *if running time is a polynomial whose degree is fixed independent of ϵ. (Can hide arbitrary constants depending only on ϵ.)*

Goal for the 00's: find efficient schemes. New framework used to obtain approximation schemes for many planar graph problems.

Planar duality: given a planar graph, form the dual by taking a node for each face and an edge for each edge. Deletion and contraction are dual to one another.

2 Framework

2.1 use of new framework for approximation schemes for planar graphs

Traveling salesman	Klein, 2005	$O(n)$
Traveling salesman on subset of vertices	Klein, 2006	$O(n \log n)$
2-edge-connected spanning subgraph	Berger, Grigni, 2007	unit weights: $O(n)$ general weights: $O(n^{f(\epsilon)})$
Steiner tree	Borradaile, Klein, Mathieu, 2008	$O(n \log n)$
2-edge-connected Steiner multisubgraph	Borradaile, Klein, 2008	$O(n \log n)$
Steiner forest	Bateni, Hajiaghayi, Marx, 2010 Eisenstat, Klein, Mathieu, 2011	$O(n^{f(\epsilon)})$ $O(n \log^3 n)$
Prize-collecting Steiner tree, TSP, stroll	Bateni, Checkuri, Ene, Haijiaghayi, Korula, Marx, 2011	$O(n^c)$ See footnote[1]
Multiterminal cut	Bateni, Hajiaghayi, Klein, Mathieu, 2011	$O(n^c)$

2.2 Framework generalized to broader graph classes

Steiner tree in bounded-genus graphs	Borradaile, Demaine, Tazari, 2009	$O(n \log n)$
Traveling salesman in excluded-minor graphs	Demaine, Hajiaghayi, Kawarabayashi, 2011	$O(n^{f(k,\epsilon)})$

2.3 Framework

- *Delete* (dually, contract) some edges while keeping OPT from increasing by more than $1 + \epsilon$ factor. Ensure total cost of resulting graph is $O(OPT)$.
- *Contract* (dually, delete) edges of total cost at most $1/p$ times total (p is a parameter). Ensure resulting graph has branchwidth $O(p)$.
- Find (near-)optimal solution in low-branchwidth graph (dynamic programming).
- Lift solution to original graph, increasing cost by $1/p \times O(OPT)$.

Canonical primal problem: traveling salesman (contracting edges only decreases the cost of the tour). Canonical dual problem: multiway cut (contracting edges only increases the cost of the cut).

Step 2 Step 2 (dual) is implicit in Baker's work. For planar graphs, do breadth-first search, p-color the levels round-robin, and delete the cheapest level. Resulting subgraphs have low branchwidth (treewidth). Step 2 (primal) run it in the dual.

Step 1 Key step: "spanner" construction. For Step 2, we need a graph of low weight so that the cost of the Step 2 edges is small. How to ensure that the resulting graph approximately preserves an optimal tour? It suffices to replace each deleted edge by some path that's not too much longer. If you can do that, then the tour is approximately preserved.

Constructing a spanner for TSP We need at least a spanning tree, so start with the minimum-weight spanning tree. Choose additional edges of total weight $\leq (2/\epsilon) weight(MST)$.

$O(n^2)$ due to Althoffer–Das–Dobkin–Joseph–Soares 1993, linear time due to Klein 2005. Here is the linear-time algorithm:

1. Let T be the minimum-weight spanning tree. Include it in the spanner.
2. "Cut" along T, duplicating edges and vertices.
3. Consider resulting face as infinite face.
4. Consider nontree edges in order. For each such edge uv, if $(1+\epsilon)$weight(uv) $\leq$weight of corresponding boundary subpath, then add uv to spanner and chop along uv. Do this repeatedly.

This algorithm is linear-time. Analysis: for each edge added to spanner, boundary weight goes down by at least ϵ weight(uv). Total weight added is at most $\epsilon^{-1} \cdot$ decrease in boundary weight. Initially, boundary is twice the weight of the MST. Total weight of spanner is $2(1+\epsilon^{-1}) weight(MST)$.

Theorem 2. *(Althoffer et al.) For any undirected planar graph G with edge-weights, $\exists$ subgraph of cost $\leq 2(1+\epsilon^{-1}) weight(MST)$ such that $\forall u, v \in V$, u-to-v distance in subgraph $\leq (1+\epsilon)$ u-to-v distance in G.*

Corollary 3. *There is a linear-time approximation scheme for traveling salesman in planar graphs.*

Traveling salesman and other problems on a subset of vertices

Theorem 4. *For any undirected planar graph G with edge-weights and any given subset S of vertices, $\exists$ subgraph of cost $\leq f(\epsilon)$min Steiner tree cost such that min Steiner tree cost in subgraph $\leq (1+\epsilon)$ min Steiner tree cost in G.*

Tool for Step 1: Brick decomposition

1. Find a 2-approximate Steiner tree T.
2. Brick decomposition is a subgraph containing T. Brick decomposition has weight O(weight of the tree). Bricks are faces of M.
3. Cut along brick boundaries. Connect bricks using a constant number $c(\epsilon)$ of portal edges.

Theorem 5. *(TSP structure theorem) There is a $1+\epsilon$-approximate tour that uses portal edges to go between bricks. (Include M and for each brick, all shortest paths between portals.)*

Theorem 6. *(Steiner structure theorem) There is a $1+\epsilon$-approximate Steiner tree that uses portal edges to go between bricks.*

Tazari and Müller-Hannemann 2009: efficient implementation of approximate Steiner-tree.

Coping with disconnected subgraphs: *prize-collecting clustering* (MohammadTaghi's talk).

Open problems

Need new techniques for

- Facility location
- Vehicle routing problems
- k-tree
- vertex-weighted Steiner tree
- directed Steiner tree
- two-edge-connected Steiner
- two-vertex-connected Steiner

Advertisement I'm writing a book about (some) optimization algorithms for planar graphs. Email me if you want to receive a draft.

Also, we are working to develop a library of reference implementations of planar-graph algorithms.