Algorithms IX

Steiner tree and TSP

Guoqiang Li

School of Software, Shanghai Jiao Tong University

Homework

- Assignment 3 is announced! (deadline May 11)

Steiner Tree

Steiner Tree

- The Steiner tree problem was defined by Gauss in a letter he wrote to Schumacher.
- The problem occupies a central place in the field of approximation algorithms.
- The problem has a wide range of applications, all the way from finding minimum length interconnection of terminals in VLSI design to constructing phylogeny trees in computational biology.
- We will present constant factor algorithms for metric Steiner tree, and the rest of the problem can be reduced to this case.

Metric Steiner Problem

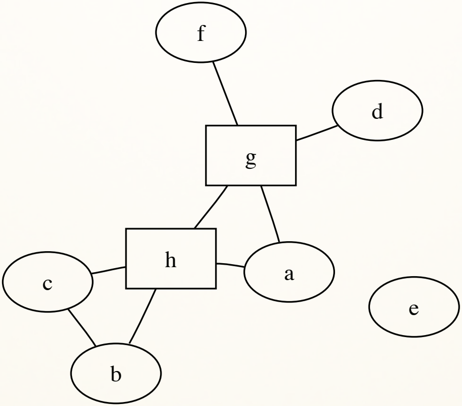
Steiner Tree

Given an undirected graph $G = (V, E)$ with nonnegative edge costs and whose vertices are partitioned into two sets, required and Steiner, find a minimum cost tree in G that contains all the required vertices and any subset of the Steiner vertices.

With a restriction to instances in which the edge costs satisfy the triangle inequality, i.e., G is a complete undirected graph, and for any three vertices u, v, and w, $cost(u, v) \leq cost(u, w) + cost(v, w)$, it is named the metric Steiner tree problem.

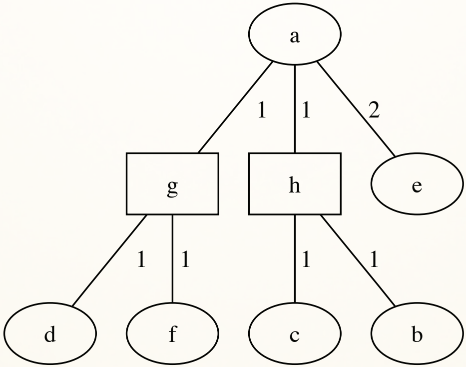
An Example

If we have a Steiner graph, with the cost 1 to the points connected by an edge, and cost 2 otherwise.



An Example

The optimal one:



Approximation Factor Preservation

Theorem

There is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.

Approximation Factor Preservation

Proof

Firstly, we will transform, in polynomial time, an instance I of the Steiner tree problem, consisting of graph $G = (V, E)$, to an instance I' of the metric Steiner tree problem.

- Let G' be the complete undirected graph on vertex set V. Define the cost of edge (u, v) in G' to be the cost of a shortest $u - v$ path in G. G' is called the metric closure of G.
- The partition of V into required and Steiner vertices in I is the same as in I.
- For any edge $(u, v) \in E$, its cost in G' is no more than its cost in G.
- Therefore, the cost of an optimal solution in I' does not exceed the cost of an optimal solution in I.

Approximation Factor Preservation

Proof

Next, given a Steiner tree T' in I', we will show how to obtain, in polynomial time a Steiner tree T in I of at most the same cost.

- The cost of an edge (u, v) in G' corresponds to the cost of a path in G.
- Replace each edge of T' by the corresponding path to obtain a subgraph of G.
- Clearly, in this subgraph, all the required vertices are connected. However, this subgraph may, in general, contain cycles.
- If so, remove edges to obtain tree T. This completes the approximation factor preserving reduction.

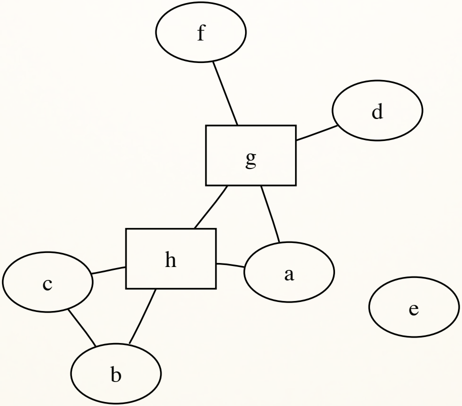
MST-based Algorithm

Algorithm

Let R denote the set of required vertices. It is easy to verify that a minimum spanning tree on R is a feasible solution for this problem.

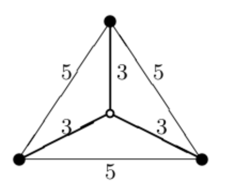
An Example

If we have a Steiner graph, with the cost 1 to the points connected by an edge, and cost 2 otherwise.



A Counterexample

The cost of MST may not be optimal:



2-approximation Algorithm

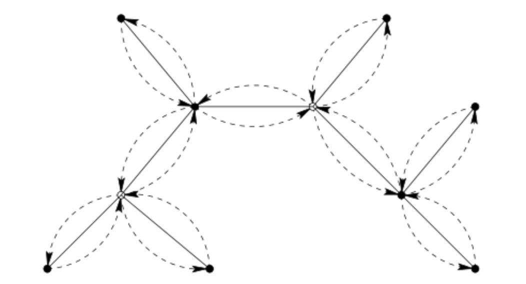
However, MST-based algorithm is a good approximation algorithm.

The cost of an MST on R is within $2 \cdot \mathrm{OPT}$.

The Analysis

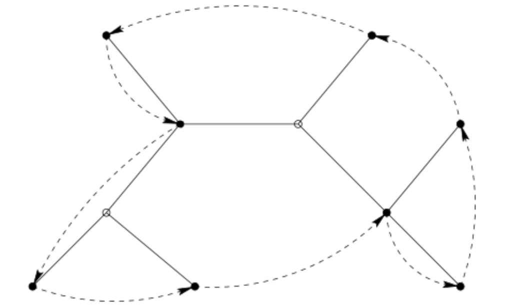
Consider a Steiner tree of cost OPT. By doubling its edges we obtain an Eulerian graph connecting all vertices of R and, possibly, some Steiner vertices.

Find an Euler tour of this graph, for example by traversing the edges in DFS (depth first search) order:



The Analysis

The cost of this Euler tour is 2·OPT. Next obtain a Rudrata cycle on the vertices of R by traversing the Euler tour and short-cutting Steiner vertices and previously visited vertices of R:

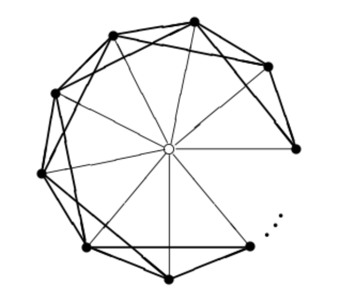


The Analysis

- Because of triangle inequality, the shortcuts do not increase the cost of the tour.
- If we delete one edge of this Rudrata cycle, we obtain a path that spans R and has cost at most $2 \cdot \mathrm{OPT}$.
- This path is also a spanning tree on R. Hence, the MST on R has cost at most $2 \cdot \mathrm{OPT}$.

Tightness of the Analysis

Consider a graph with n required vertices and one Steiner vertex. An edge between the Steiner vertex and a required vertex has cost 1, and an edge between two required vertices has cost 2:



Metric TSP

Traveling Salesman Problem

Given a complete graph with nonnegative edge costs, find a minimum cost cycle visiting every vertex exactly once.

- Interestingly, TSP cannot be approximated within any polynomial bounded ratio.
- For any polynomial time computable function $\alpha(n)$, TSP cannot be approximated within a factor of $\alpha(n)$, unless **P = NP**.

Rudrata cycle $\rightarrow$ TSP

- Given a graph $G = (V, E)$, construct the following `instance` of the TSP:
  - The set of cities is the same as V.
  - The distance between cities u and v is 1 if $\{u, v\}$ is an edge of G and $1 + \alpha$ otherwise, for some $\alpha > 1$ to be determined.
  - The budget of the TSP `instance` is equal to the number of nodes, $|V|$.
- If G has a Rudrata cycle, then the same cycle is also a tour within the budget of the TSP `instance`.
- Conversely, if G has no Rudrata cycle, then there is no solution: the cheapest possible TSP tour has cost at least $n + \alpha$ (it must use at least one edge of length $1 + \alpha$, and the total length of all $n - 1$ others is at least $n - 1$).
- Thus Rudrata Cycle reduces to TSP.

Proof of the Inapproximability

We will describe a reduction from Rudrata cycle problem (which is **NP**-completeness) to TSP problem.

That is, transform a graph G on n vertices to an edge-weighted complete graph G' on n vertices such that

- if G has a Rudrata cycle, then the cost of an optimal TSP tour in G is n, and
- if G does not have a Rudrata cycle, then an optimal TSP tour in G is of cost $> \alpha(n) \cdot n$.

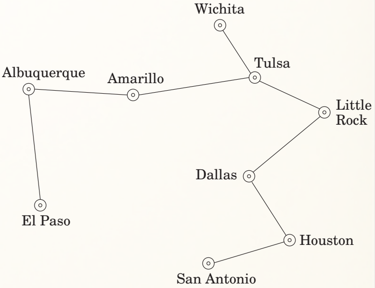
Assign a weight of 1 to edges of G, and a weight of $\alpha(n) \cdot n$ to nonedges, to obtain G'.

Metric TSP

- Notice that in order to obtain such a strong nonapproximability result, we had to assign edge costs that violate triangle inequality.
- If we restrict ourselves to graphs in which edge costs satisfy triangle inequality, i.e., consider metric TSP, the problem remains NP-complete, but it is no longer hard to approximate.

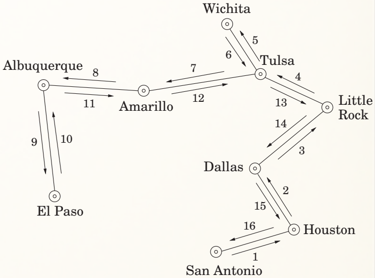
TSP on Metric Space

- Removing any edge from a traveling salesman tour leaves a path through all the vertices, which is a spanning tree.
- Therefore,
  Tsp cost $\geq$ cost of this path $\geq$ Mst cost



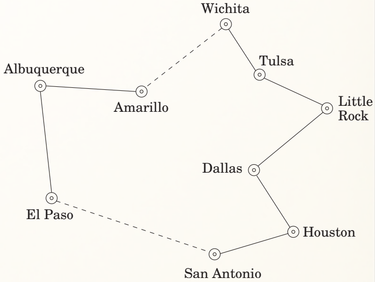
TSP on Metric Space

- If we can use each edge twice, then by following the shape of the Mst we end up with a tour that visits all the cities, some of them more than once.



TSP on Metric Space

- To fix the problem, the tour should simply skip any city it is about to revisit, and instead move directly to the next new city in its list.
- By the triangle inequality, these bypasses can only make the overall tour shorter.



A Simple Factor 2 Algorithm

Consider the following algorithm:

1. Find an MST, T of G
2. Double every edge of the MST to obtain an Eulerian graph.
3. Find an Eulerian tour, $\mathcal{T}$, on this graph.
4. Output the tour that visits vertices of G in the order of their first appearance in $\mathcal{T}$. Let $\mathcal{C}$ be this tour.

The above algorithm is a factor 2 approximation algorithm for metric TSP.

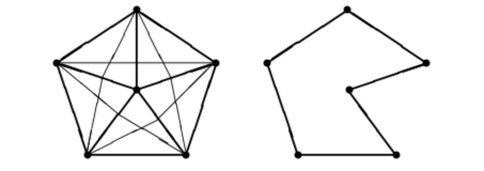
Analysis

- $cost(T) \leq$OPT.
- Since $\mathcal{T}$ contains each edge of T twice, $\text{cost}(\mathcal{T}) = 2 \cdot \text{cost}(T)$.
- Because of triangle inequality, after the short-cutting (step 4) step, $\text{cost}(\mathcal{C}) \leq \text{cost}(\mathcal{T})$.
- Combining these inequalities we get that

$$\text{cost}(\mathcal{C}) \leq 2 \cdot \text{OPT}.$$

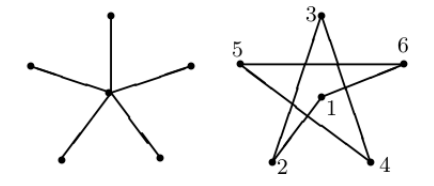
A Tight Example

- A tight example for this algorithm is given by a complete graph on n vertices with edges of cost 1 and 2.
- We present the graph for $n = 6$ below, where thick edges have cost 1 and remaining edges have cost 2.
- For arbitrary n the graph has $2n - 2$ edges of cost 1, with these edges forming the union of a star and an $n - 1$ cycle; all remaining edges have cost 2.
- The optimal TSP tour has cost n, as shown below for $n = 6$.



A Tight Example

Suppose that the MST found by the algorithm is the spanning star created by edges of cost 1. Moreover, suppose that the Euler tour constructed in Step 3 visits vertices in order shown below for $n = 6$:

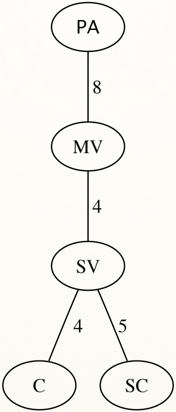


Then the tour obtained after short-cutting contains $n - 2$ edges of cost 2 and has a total cost of $2n - 2$. Asymptotically, this is twice the cost of the optimal TSP tour.

An Example

	C	MV	PA	SC	SV
C	0	7	12	7	4
MV		0	8	9	4
PA			0	14	10
SC				0	5
SV					0

An Example



Improving the factor to 3/2

- Is there a cheaper Euler tour than that found by doubling an MST?
- Recall that a graph has an Euler tour iff all its vertices have even degrees.
- Thus, we only need to be concerned about the vertices of odd degree in the MST.

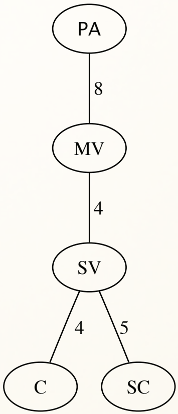
Let V' denote the set of vertices of odd degree in the MST. Now, if we add to the MST a minimum cost perfect matching on V, every vertex will have an even degree, and we get an Eulerian graph.

The Algorithm

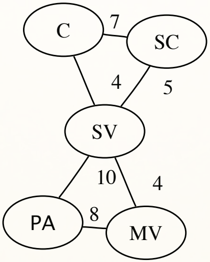
1. Find an MST of G, say T.
2. Compute a minimum cost perfect matching, M, on the set of odd-degree vertices of T. Add M to T and obtain an Eulerian graph.
3. Find an Euler tour, $\mathcal{T}$, of this graph.
4. Output the tour that visits vertices of G in order of their first appearance in $\mathcal{T}$. Let $\mathcal{C}$ be this tour.

An Example

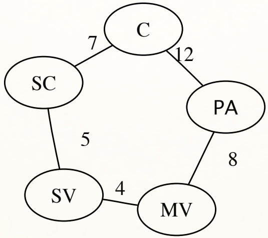
	C	MV	PA	SC	SV
C	0	7	12	7	4
MV		0	8	9	4
PA			0	14	10
SC				0	5
SV					0



An Example



An Example



Analysis

Lemma

Let $V' \subseteq V$, such that $|V|$ is even, and let M be a minimum cost perfect matching on V. Then, $\text{cost}(M) \leq \text{OPT}/2$.

Consider an optimal TSP tour of G, say τ.

- Let τ' be the tour on V' obtained by short-cutting τ. By the triangle inequality, $\text{cost}(\tau') \leq \text{cost}(\tau)$.
- Now, τ' is the union of two perfect matchings on V', each consisting of alternate edges of τ.
- Thus, the cheaper of these matchings has cost $\leq \text{cost}(\tau')/2 \leq \text{OPT}/2$. Hence the optimal matching also has cost at most $\text{OPT}/2$.

Analysis (cont'd)

Theorem
The above algorithm achieves an approximation guarantee of $3/2$ for metric TSP.

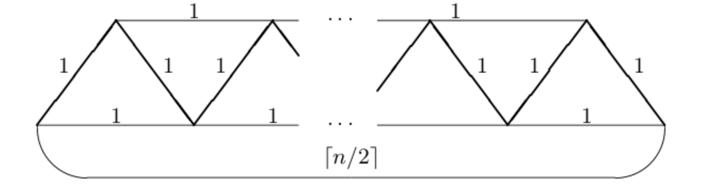
The cost of the Euler tour,

$$\text{cost}(\mathcal{T}) \le \text{cost}(T) + \text{cost}(M) \le \text{OPT} + \frac{1}{2}\text{OPT} = \frac{3}{2}\text{OPT}$$

Using the triangle inequality, $\text{cost}(\mathcal{C}) \le \text{cost}(\mathcal{T})$, and the theorem follows.

A Tight Example

A tight example for this algorithm is given by the following graph on n vertices, with n odd.



Thick edges represent the MST found in step 1. This MST has only two odd vertices, and by adding the edge joining them we obtain a traveling salesman tour of cost $(n-1)+n/2$. In contrast, the optimal tour has cost n.

A Better Algorithm?

Finding a better approximation algorithm for metric TSP is currently one of the outstanding open problems in this area. Many researchers have conjectured that an approximation factor of $4/3$ may be achievable.

Referred Materials

- Content of this lecture comes from Chapter 3 in [Vaz04].