Approximation Algorithms

Metric Steiner Tree

Kaouri Georgia

NTUA, Spring Semester 2009

Steiner Tree problem

- Given an undirected graph G=(V,E) with non negative edge costs and whose vertices are partitioned into two sets, required (denoted by R) and Steiner (denoted by S)
- Find a minimum cost tree in G containing all vertices in R and any subset of vertices in S

metric Steiner Tree problem

- Additional restriction: the graph is complete and the edge costs satisfy the triangle inequality, i.e. for any three vertices u,v,w in V

$$cost(u,v) \leq cost(u,w) + cost(v,w)$$

Approximation Factor Preserving Reduction from Steiner Tree problem to metric Steiner Tree problem

- **Theorem:** Any a(n)-approximation algorithm for metric Steiner Tree can be transformed to an a(n)-approximation algorithm for Steiner Tree

- **Proof:** We will transform an instance I of the Steiner Tree problem to an instance I' of the metric Steiner Tree problem.

  We construct a complete undirected graph G' on vertex set V. We keep R' and S' as in I. The cost between vertices u', v' in G' is equal to the cost of the shortest u-v path in G (G' is the metric closure of G).

  OPT(I')≤OPT(I), since cost(u',v') ≤cost(u,v)

  Given a Steiner Tree T' in I', we will construct a Steiner Tree in I by replacing each edge of T' by its corresponding path in G and deleting the edges that create cycles.

  - all the required vertices are connected

$$SOL(I) \le SOL(I') \le a(n)OPT(I') \le a(n)OPT(I)$$

2-approximation algorithm for metric Steiner Tree problem

- **Note:** Any minimum spanning tree (MST) on R is a feasible solution for metric Steiner Tree problem
- **Algorithm:** Find a MST on R.
- **Theorem:** cost(MST on R) $\leq$ 2 OPT.
- **Proof:** Technique similar to TSP. Consider a Steiner tree of cost OPT. Double the edges to obtain an Euler graph (its cost is 2 OPT). Find an Euler graph and obtain a Hamilton cycle by “shortcutting” Steiner vertices and visited vertices. Delete one edge of the HC to obtain a tree that spans R.

  The shortcuts do not increase the cost of the solution produced. Therefore, this solution has cost at most 2 OPT.

Tight example

- Consider a graph with one central Steiner vertex u and n peripheral required vertices. The cost of the edges connecting u with the n nodes is 1 and the cost of the edges between the n nodes is 2.

- The MST algorithm produces a solution with cost 2(n-1), when the optimal cost is n.