Approximation Algorithms

Homework #2

Olga Ohrimenko

09/19/2008

Exercise 3.1

Given a graph $G = (V, E)$, a required set of vertices R and optimal subset C of Steiner vertices that need to be included in the tree, optimal Steiner tree of cost OPT can be built in polynomial time.

Algorithm:

Remove each vertex $v \in V$ such that $v \notin R$, $v \notin C$ and edges that connect it to G to obtain graph $G' = (R \cup C, E')$, where $E' \subseteq E$. For G' find a minimum spanning tree (MST) T and output it.

Claim: T is the optimal Steiner tree on R in G.

Proof: T a Steiner tree as all vertices in R are present in T. By the definition of MST, T is a spanning tree with minimum sum of edge costs that connects all vertices in T. As we know that optimal Steiner tree contains all vertices in C, MST on the union of R and C will be the optimal Steiner tree.

Analysis: Removing vertices that do not belong to $R \cup C$ and their edges can be done in polynomial time by considering each vertex in V. Greedy algorithms for building MST run in polynomial time. Thus algorithm described above finds an optimal Steiner tree and runs in polynomial time. □