Saarland Informatics Campus

Assignment 1, Approximation Algorithms
Summer term 2017

Tobias Mömke, Hang Zhou
`http://www-cc.cs.uni-saarland.de/course/61/`

Due: 2 May 2017

Exercise 1.1 (10 Points) Let $G = (V, E, w)$ be a complete graph with vertices V, edges E and edge weights $w\colon E \to \mathbb{R}^+$. Suppose that w satisfies the triangle inequality $w(\{u, v\}) \leq w(\{u, w\}) + w(\{w, v\})$ for each combination of distinct vertices u, v, w. (i. e., G is a metric graph). Let $s, t \in V$ be two vertices. The problem s, t-path-TSP is to find a minimum cost path from s to t in G such that each vertex of V is visited exactly once.

(a) Show that s, t-path-TSP is NP-hard.

(b) Design a 2-approximation algorithm for s, t-path-TSP. (Of course, you have to show that the algorithm has the desired properties, in particular that the approximation ratio is indeed 2.)

Exercise 1.2 (10 Points) Let T be a tree. Show formally that

(a) in a tour determined by a DFS search, each edge of T is traversed exactly twice; and

(b) in a metric graph G, an Eulerian tour can be transformed into a Hamiltonian cycle of at most the same cost.

Exercise 1.3 (10 Points) Let $G = (V, E, w)$ be a complete graph with edge weights $w\colon E \to \{1, 2\}$.

(a) Show that G is a metric graph.

(b) Consider the following algorithm (assuming that $|V|$ is even).

- Compute a minimum cost perfect matching M of G. Each edge in M forms a component.
- As long as there are at least two components C_1, C_2, add an edge between them to connect them to a path. (The components have two degree one vertices each, therefore there are four edges that we can choose to connect the two components. The algorithm simply picks the first one it sees.)
- If there is only one component left, add an edge between its ends (to connect the path to a cycle).

Show that the algorithm is a 1.5-approximation algorithm for TSP with edge weights one and two.

Exercise 1.4 (10 Points) Let G be a metric graph (as in Exercise 1). A degree-3-bounded spanning tree of G is a spanning tree where each vertex has a degree of at most 3.

(a) Design a 2-approximation algorithm for the problem to find a minimum-cost degree-3-bounded spanning tree. (Of course, you have to show that the algorithm has the desired properties, in particular that the approximation ratio is indeed 2.)

(b) Let k be a parameter of the input. Show that the problem to find a minimum cost degree-k-bounded spanning tree is NP-hard.

(c) Show that the degree-3-bounded spanning tree problem is NP-hard.