Abteilung Algorithmik
Institut für Betriebssysteme
und Rechnerverbund
TU Braunschweig

SS 12

Prof. Dr. Sándor Fekete
Dr. Christiane Schmidt

Approximation Algorithms
Homework Set 0, 25. 04. 2012

Solutions to this homework set will not be evaluated, the homework set is treated in the first small tutorial.

Exercise 1 (The Metric Traveling Salesman Problem):

(a) In the first tutorial on April 26th we consider an idea for constructing a tour via doubling a MST, and using short cuts if possible (using the triangle inequality).

Show that this procedure gives an approximation algorithm and determine its approximation factor.

(b) Give an example to show that your bound for the algorithm is tight, i.e., give an example in which your given approximation factor is achieved.

Exercise 2 (The Traveling Salesman Problem):
We consider the general—not the geometric (or metric)—variant of the Traveling Salesman Problem.

(a) Show that the TSP cannot be approximated within a constant factor, i.e., show that there is no algorithm with constant approximation factor δ (unless $P = NP$).

For this proof you may use the Hamiltonian circuit problem (HCP). The input of the HCP is a graph G (not necessarily complete) with vertex set V and edge set E. The decision problem is the following: does a circuit in G exist that visits each vertex exactly once?

For the required proof we consider an arbitrary instance of HCP and define $c_e = 1$ if $e \in E$ and $c_e = n(\delta + 1)$ if $e \notin E$. Thus, we have an instance of the TSP. Let $|ALG|$ be the value of ALG. Deduce that with this algorithm it would be possible to decide HCP. (Hint: either $|ALG| \leq \delta n$ or $|ALG| > \delta n$.)

(b) Consider the following greedy algorithm for the TSP in complete graphs: Let S denote the set of all visited vertices (at the start $S := \emptyset$). Start with an arbitrary vertx $v \in V$. Add v to V and choose an edge $e = \{v, w\}$ of minimal weight, with $w \in V \setminus S$. Proceed with w. In case $S = V$, move back to the start vertex.

Give an example in which the ratio of algorithm to optimum can be arbitrarily bad.