Thomas Rothvoß
Location: ELD120
Discussion: 28.04.10

Exercises

Approximation Algorithms

Spring 2010

Sheet 8

Exercise 1

For the EUCLIDEAN k-TSP problem, we are given points $v_1,\dots,v_n \in \mathbb{Q}^2$ in the plane and a parameter $k \in \{1,\dots,n\}$. The goal is to find a minimum length tour, visiting *at least k nodes*. Here the length is measured using the Euclidean distances. Give a PTAS for this problem (by adapting Arora's algorithm).

Solution:

The first difference lies in the fact that $OPT \geq L$ does not necessarily hold. Let π^* be the cheapest tour, visiting k cities. But we can guess the leftmost, rightmost, lowest and highest visited city (there are $\leq n^4$ combinations). Then we delete the cities, which are not within the square, spanned by these 4 cities. Then $OPT \geq L$ holds (after an appropriate translation/scaling).

We use the same definition for well rounded tours as Arora in the TSP algorithm. We apply the structure theorem to π^* and obtain a well-rounded tour visiting the same set of k cities of cost $(1+O(\varepsilon))OPT$. Next, we argue, how to find it.

We use table entries $A(Q,k',(s_1,t_1),\dots,(s_q,t_q))$ for every square Q, $k' \in \{0,\dots,k\}$, $q \leq 4/\varepsilon$ and portals s_i,t_i of Q. Here $A(Q,k',(s_1,t_1),\dots,(s_q,t_q))$ denotes the cost of the cheapest extension of q subtours to a well-rounded tour, where ith subtour goes from s_i to t_i and k' many nodes are visited inside Q. We can compute such entries again bottom-up. Let $Q_1,\dots,Q_4$ be the subsquares of Q. Guess the visited portals of the Q_j's and the numbers $k_j \in \mathbb{N}_0$, how many nodes are visited in Q_j (and $k_1+\ldots+k_4 = k'$). Look up and sum up the cost for the according subtour extensions in the subsquares.

Exercise 2

For EUCLIDEAN STEINER TREE, we are given *terminals* $v_1,\dots,v_n \in \mathbb{Q}^2$ in the plane. The goal is to find the cheapest Steiner tree T, spanning all terminals. Here the cost of the tree is measured using the Euclidean distances. For the Steiner tree T one is allowed to add arbitrary points from $\mathbb{Q}^2$ as Steiner nodes in order to make the tree cheaper. Give a PTAS for this problem.

Hint: If might be helpful to answer the following questions.

i) Argue, that the discretization still costs $O(\varepsilon)\cdot OPT$.

ii) Which properties should a *well-rounded* Steiner tree have?

iii) How could suitable table entries for the dynamic program look like? How can you compute them?

iv) How would the patching lemma be for Steiner trees?

v) What about the structure theorem for Steiner tree?

Solution:

i) We compute a bounding box as for TSP. Clearly we have to connect also the terminals that have distance $L/2$, hence $OPT \geq L/2$. Next, any Steiner node must have degree ≥ 3, hence the Steiner tree contains $\leq 2n$ edges. The discretization changes the cost by at most $2n \cdot 2 \leq 8\varepsilon \cdot OPT$.

ii) We call a Steiner tree T *well-rounded*, if it crosses each square only in portals and the number of crossings per square are at most $4/\varepsilon$.

iii) If we intersect a tree with a square Q in the dissection, we obtain a forest. Hence, let $P_1, \dots, P_q$ be (disjoint) subsets of the portals of Q. Then we want

$$A(Q, P_1, \dots, P_q) = \text{cost of the cheapest forest consisting of trees } T_1, \dots, T_q \text{ inside } Q, \text{such that } T_i \text{ spans the portals } P_i. \text{ Furthermore each node inside } Q \text{ must be connected to some of these trees.}$$

Similar to TSP, we consider the subsquares $Q_1, \dots, Q_4$ and guess how the optimum well-rounded forest would cross the Q_i's. Then we read the cost for the corresponding subforest from the table and sum up their cost.

If Q is an empty square, the cheapest steiner forest, connecting each P_i can be easily computed since $|P_i| = O(1)$.

iv) The patching lemma for a Steiner tree T is actually even simpler: We consider a line segment ℓ of length s, which is crossed by T an arbitrary number of times. We cut T at ℓ. Then we add line segments of length s each to the left and right and add a cost 0 edge, connecting both. This reduces the crossings to 1 and costs $2s$.

v) For the structure theorem: We still have $OPT = \Theta(1) \cdot \#\text{crossings}$. Bending through portals works exactly as for TSP. The `MODIFY` procedure only needs a working patching lemma (which we have, see iv)).