Research Institute for Discrete Mathematics
Chip Design
Summer Term 2014

Prof. Dr. S. Hougardy
R. Scheifele, M. Sc.

Exercise Set 4

Exercise 4.1:
For a finite set $V \subseteq \mathbb{R}^2$, the ℓ_1-Voronoi diagram consists of the regions

$$P_v := \left\{ x \in \mathbb{R}^2 \ : \ ||x - v||_1 = \min_{w \in V} ||x - w||_1 \right\}$$

for $v \in V$. The ℓ_1-Delaunay triangulation of V is the graph

$$(V, \ \{\{v, w\} \subseteq V, \ v \neq w, \ |P_v \cap P_w| > 1\}).$$

Assume that the slope of each straight line connecting two elements of V is neither 1 nor -1.

a) Show that the ℓ_1-Delaunay triangulation is a planar graph.

b) Show how a rectilinear minimum spanning tree for V can be computed in $\mathcal{O}(|V| \log |V|)$ time. You can use the fact that the Delaunay triangulation can be computed in $O(|V| \log |V|)$ time.

c) Show that the ℓ_1-Delaunay triangulation is not necessarily planar without the requirement that the slope of each straight line connecting two elements of V is neither 1 nor -1.

(2 + 3 + 2 points)

Exercise 4.2:
Let N be an instance of the Rectilinear Steiner Tree Problem and $r \in N$. For a rectilinear Steiner tree Y we denote by $f(Y)$ the maximum length of a path from r to any $N \backslash r$ in Y.

a) Describe an instance in which no shortest Steiner tree minimizes $f(Y)$ and no Steiner tree minimizing $f(Y)$ is shortest.

b) Consider the problem of finding a shortest Steiner tree Y minimizing $f(Y)$ among all shortest Steiner trees. Is there always a tree with these properties which is a subgraph of the Hanan grid?

(1+3 points)

Exercise 4.3:
Given a complete graph $G = (V, E)$ with metric edge costs $dist : V \times V \to \mathbb{R}_{\geq 0}$ and a spanning arborescence Y_0 with root s (i.e. $V(Y_0) = V(G)$) and $\varepsilon > 0$, consider the following algorithm connstructing a spanning arborescence Y with root s:

1. Start with $Y = Y_0$.
2. Traverse the edges of Y_0 in depth-first search order, i.e. every edge is traversed twice.
3. If $e = (v, w)$ is traversed for the first time: Check if $dist_Y(s, w) > (1+\varepsilon) \cdot dist(s, w)$. If true, delete the edge (v, w) from Y and add the edge (s, w) instead.
4. If $e = (v, w)$ is traversed for the second time: Check if $dist_Y(s, v) > dist_Y(s, w) + dist(v, w)$. If true, delete the incoming edge of v from Y and add the edge (w, v) instead.

Here $dist_Y(x, y)$ denotes the length of the x-y path in Y w.r.t. to $dist$ for $x, y \in V$.
Prove: The above algorithm computes a spanning arborescence Y with

- $dist_Y(s, v) \leq (1 + \varepsilon) \cdot dist(s, v)$ and
- $\sum_{(v,w)\in E(Y)} dist(v, w) \leq (1 + \frac{2}{\varepsilon}) \sum_{(v,w)\in E(Y_0)} dist(v, w)$

and can be implemented in $\mathcal{O}(|V(Y_0)|)$ time.

(5 points)

Deadline: Thursday, May 8, before the lecture.
The websites for lecture and exercises are linked at

http://www.or.uni-bonn.de/lectures/ss14/ss14.html

In case of any questions feel free to contact me at scheifele@or.uni-bonn.de .