Advanced algorithms

Exercise sheet #6 – Approximation algorithms

November 9, 2022

Exercice 1 (TSP with triangle inequality)**.** Let G be a complete graph with n vertices, labelled from 1 to n. To each of the $\frac{1}{2}n(n-1)$ edges (u, v) is associated a distance $d(u, v)$. The traveling salesman looks for a minimum-length tour that starts and ends on 1 and visits every vertex exactly once. The decision version of this problem is NP-complete.

We assume furthermore that the distance d satisfies the triangle inequality: $d(u, v) \leqslant d(u, w) + d(w, v)$ for any vertex u, v and w.

Let T be a minimum spanning tree, rooted at 1, and let H be the tour obtained by the pre-order depth first traversal of T.

(a) What is the complexity of computing H?

(b) Let H^* be an optimal tour. Show that the length $c(T)$ of T (that is the sum of the distances of the edges in T) is at most the length $c(H^*)$ of H^*.

(c) Using the triangle inequality, show that $c(H) \leqslant 2c(T)$. Deduce an approximation algorithm, with approximation factor 2, for computing an optimal tour.

Exercice 2 (Multiterminal cut (Pâle 2013))**.** Let $G = (V, E)$ be a connected graph endowed with a weight function $c(e) \geq 0$ for each edge $e \in E$ and with a distinguished subset S of vertices, called *terminals*.

A *multiterminal cut* of G is a set of edges $F \subseteq E$ whose removal would disconnect all terminals from each other.

The weight of a multiterminal cut is the sum of the weight of its elements. Given S, we aim at computing a minimum-weight multiterminal cut, or rather an approximation.

(a) Given a multiterminal cut F and $v \in S$, let $G_v[F]$ be the connected component of $G \setminus F$ containing v. Moreover, let F_v be the subset of F of all edges with exactly one end in $G_v[F]$. Show that any path in G from v to any other $w \in S$ has an edge in F_v.

(b) For $v \in S$, let E_v be a minimum-weight set of edges such that any path in G from v to any other $w \in S$ has an edge in E_v. Show that E_v can be computed in polynomial time. What is the complexity of your algorithm ?

(c) Deduce a 2-approximation algorithm for the problem of computing a minimum-weight multiterminal cut.

Exercice 3 (Vertex cover with linear programming)**.** Let $G = (V, E)$ be a graph with a weight function $c(v) \geq 0$ on the vertices. We aim at computing an approximate minimum-weight vertex cover of G. Recall that a vertex cover is a set $S \subset V$ so that each edge has at least one end in S.

Consider the following linear program:

$$\begin{aligned} \text{minimize } & \sum_{v \in V} c(v) x_v \\ \text{such that } & x_u + x_v \geqslant 1, \quad \forall \{u, v\} \in E \\ & 1 \geq x_v \geq 0, \quad \forall v \in V, \end{aligned}$$

with the optimal value λ^* and an optimal solution $(x_v^*)_{v \in V}$.

(a) Let S^* be a minimum-weight vertex cover of G. Show that $c(S^*) \geqslant \lambda^*$.

(b) From the optimal solution $(x_v^*)_{v \in V}$, construct a vertex cover S of G such that $c(S) \leqslant 2\lambda^*$.

Exercice 4 (The center selection problem)**.** Let V be a finite set endowed with a distance function $d : V \times V \to [0, \infty)$ which satisfies the usual properties of distance functions:

- Separation: $d(u, v) = 0 \Leftrightarrow u = v$ for all $u, v \in S$,
- Symmetry: $d(u, v) = d(v, u)$ for all $u, v \in S$,
- Triangle inequality: $d(u, v) \leq d(u, w) + d(w, v)$ for all $u, v, w \in S$.

For any subset $S \subset V$, we define its covering radius

$$\operatorname{rad}(S) = \max_{v \in V} \min_{s \in S} d(v, s).$$

It is the maximal distance of an element of V to the closest element of S. Given an integer k, a subset S of size $\leq k$ of minimal covering radius is called a set of *centers.*

(a) Let $r \geqslant 0$ and assume that there exists a subset $S^* \subseteq V$ of k centers such that $\operatorname{rad}(S^*) \leqslant r$. Design a greedy algorithm to compute a $S \subseteq V$ with $\#S \leqslant k$ and $\operatorname{rad}(S) \leqslant 2r$.

(b) Let r^* be the minimum value of $\operatorname{rad}(S^*)$, for $S^* \subseteq V$ and $\#S^* = k$. Design an algorithm to compute in polynomial time a $S \subseteq V$ with $\#S \leqslant k$ and $\operatorname{rad}(S) \leqslant 2r^*$.