Weekplan: Approximation Algorithms II

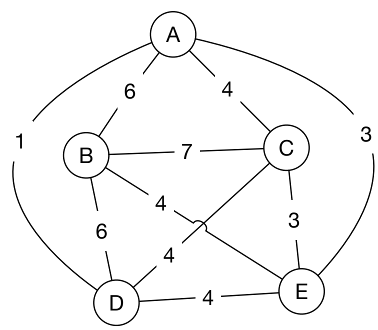
Inge Li Gørtz

References and Reading

[1] The Design of Approximation Algorithms, Williamson and Shmoys, Cambridge Press, section 2.4.

We reccomend that you read [1] in detail.

1 [w] **TSP** Run both the double tree algorithm and Christofides' algorithm on the example below. Show MST (and the matching in Christofides') and write down both τ and τ'.



2 TSP with fixed start and endpoint In the traveling salesman problem with fixed start and endpoint we are given a start point s and an endpoint $t \neq s$. The salesman must still visit each city exactly once, but he now has to start in s and end in t.

2.1 Give a 2-approximation algorithm for this problem.

2.2 [$*$] Give an algorithm with an approximation factor smaller than 2.

3 Minimum Steiner Tree In the *minimum Steiner tree* problem, we are given a complete, undirected graph $G = (V, E)$ with nonnegative edge costs $c_{ij} \geq 0$ for all edge $(i, j) \in E$. The vertices are partitioned into two sets, *required* R and *Steiner* S, and the goal is to find a minimum cost tree that contains all the required vertices and any subset of the Steiner vertices.

3.1 Assume that the edge costs satisfy the triangle inequality, i.e., G is a complete undirected graph, and for any three vertices i, j, and k, $c_{ij} \leq c_{ik} + c_{kj}$. We call this the metric Steiner tree problem. Consider the following algorithm: Compute a minimum spanning tree on the set of required vertices R. That is, compute a minimum spanning tree on the graph $G[R]$ induced on the set of required vertices R[1]. Prove that this is a 2-approximation algorithm.

3.2 Give an example showing that a minimum spanning tree on R can be bigger than the optimal Steiner tree.

[$*$] See if you can find a tight example for the algorithm, i.e., give an example showing that the minimum spanning tree on R can be almost 2 times bigger than the optimal metric Steiner tree on a graph with n vertices.

[1] $G[R]$ contains R and all edges from G that has both endpoints in R.

3.3 Now suppose that the edge costs in $G = (V, E)$ do not obey the triangle inequality, and that the input graph is connected but not necessarily complete. In this exercises we will show that there is an approximation factor preserving reduction from the Steiner tree problem to the metric Steiner tree problem.

Let G' be the complete graph on vertex set V, and define the edge costs c'_{ij} to be the cost of the shortest path from i to j in G. This is called the *metric completion* of G. Prove that the cost of an optimal Steiner tree in G is the same as the cost of an optimal Steiner tree in G'.

4 Bottleneck TSP In the metric bottleneck travelling salesman problem we have a complete graph with distances satisfying the triangle inequality, and we want to find a hamiltonian cycle such that the cost of the most costly edge in the cycle is minimized. The goal of this exercise is to give a 3-approximation algorithm for this problem.

4.1 A bottleneck minimum spanning tree of a graph G is a spanning tree minimizing the heaviest edge used. Argue that it is possible to find an optimal bottleneck MST in polynomial time.

4.2 Show that it is possible to construct a walk visiting all nodes in a bottleneck MST exactly once without shortcutting more than 2 consecutive nodes.

Hint: Show by induction that you can list the vertices of a rooted tree T in a sequence such that the number of edges in T between two adjacent vertices (including the first and the last) is at most 3, and *furthermore the vertex following the root r is a child of r.*

4.3 Give a 3-approximation algorithm for bottleneck TSP.

5 Priority k-center Consider the following variant of the k-center problem, where the vertices have priorities: Each vertex have a weight, and we want to find a set of k centers so that the maximum *prioritized* distance of a vertex to its closest center is minimized. Formally, in the *prioritized k-center problem* we are given a complete graph $G = (V, E)$ with a cost function on the edges $d : E \to Q^+$ satisfying the triangle inequality, a priority function on vertices: $p : V \to \mathbb{R}^+$, and a positive integer k. The problem is to find a set of centers $C \subseteq V$ with $|C| \leq k$ minimizing

$$r(C) = \max_{v \in V} p(v) \cdot d(v, C),$$

where

$$d(v, C) = \min_{u \in C} d(v, u).$$

The following algorithm for the prioritized k-center problem assumes we know the optimal radius r.

Algorithm 1 kCenter (G, r)

```
Set S = V and C = ∅.
while S ≠ ∅ do
    Select the heaviest vertex v ∈ S (the vertex with highest priority)
    Set C = C ∪ {v}
    Remove all vertices u from S with p(u) · d(u, v) ≤ 2r from v.
end while
Return C
```

5.1 Assume we know the optimum covering radius r. Let C be the set of centers computed by the algorithm kCenter(G, r), and let C^* be the set of optimal centers (each vertex is assigned to its closest center).

(a) Consider an iteration of the algorithm and let v be the vertex chosen in this iteration. Let c^* be the center v is assigned to in the optimal solution. Let $z \in S$ be a vertex assigned to c^* in the optimal solution. Show that $p(z) \cdot d(z, v) \leq 2r$.

(b) Show that at most one vertex from each cluster from C^* belongs to C.

5.2 Prove that Algorithm 1 is a 2-approximation algorithm for the prioritized k-center problem (assuming that we know the optimal covering radius r).

5.3 Show how to get rid of the assumption that we know the optimal covering radius.