Weekplan: Approximation Algorithms II

Inge Li Gørtz

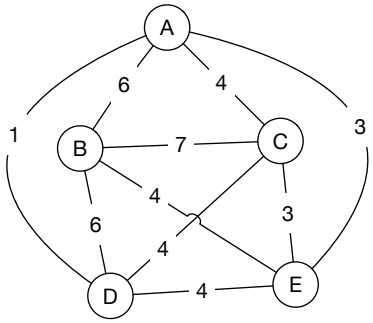
References and Reading

[1] Algorithm Design, Kleinberg and Tardos, Addison-Wesley, section 11.3.

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

We reccomend that you read both [1] and [2] 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 Consultant You are a consultant for a company that has a number of servers and need to schedule batches of jobs. Once a batch of n jobs arrives they need to be allocated to servers. The company has two types of servers: k fast servers and m slow servers. Each job i takes time t_i to process on a slow server, and time $t_i/3$ to process on a fast server. The goal is to minimize the makespan of the schedule.

You suggest that they use the simple greedy algorithm: Process jobs in any order. Assign next job on list to machine with smallest current load.

3.1 Give an example showing that this algorithm is not a 3-approximation algorithm.

3.2 Prove that this is a 4-approximation algorithm.

3.3 Give a better approximation algorithm for the problem.

4 Representative set of proteins (exercise 2 in [1]) At a lecture in a computational biology conference one of us attended a few years ago, a well-known protein chemist talked about the idea of building a "representative set" for a large collection of protein molecules whose properties we don't understand. The idea would be to intensively study the proteins in the representative set and thereby learn (by inference) about all the proteins in the full collection. To be useful, the representative set must have two properties.

- It should be relatively small, so that it will not be too expensive to study it.
- Every protein in the full collection should be "similar" to some protein in the representative set. (In this way, it truly provides some information about all the proteins.)

More concretely, there is a large set P of proteins. We define similarity on proteins by a distance function d: Given two proteins p and q, it returns a number $d(p,q) \geq 0$. In fact, the function $d(\cdot,\cdot)$ most typically used is the so-called sequence alignment measure. We will assume that the distance function satisfy the triangle inequality. There is a predefined distance cut-off Δ that's specified as part of the input to the problem; two proteins p and q are deemed to be "similar"to one another if and only if $d(p,q) \leq \Delta$. We say that a subset of P is a *representative set* if, for every protein p, there is a protein q in the subset that is similar to it—that is, for which $d(p,q) \leq \Delta$. Our goal is to find a representative set that is as small as possible.

4.1 Give a polynomial-time algorithm that approximates the minimum representative set to within a factor of $O(\log n)$. Specifically, your algorithm should have the following property: If the minimum possible size of a representative set is s^*, your algorithm should return a representative set of size at most $O(s^* \log n)$.

4.2 Note the close similarity between this problem and the Center Selection Problem—a problem for which we considered approximation algorithms in Section 11.2. Why doesn't the algorithm described there solve the current problem?

5 Vertex cover A *vertex cover* in a graph $G = (V,E)$ is a subset of vertices $S \subseteq V$ so that each edge has at least one end in S. In the *cardinality vertex cover problem* the goal is to find a vertex cover of the input graph $G = (V,E)$ of minimum size.

A *matching* in a graph $G = (V,E)$ is subset of edges $M \subseteq E$ so that no two edges of M hare an endpoint. A *maximal matching* is a matching that is maximal under inclusion. That is, adding any edge from E to the maximal matching will cause two edges in M to share an endpoint.

5.1 Show that a maximal matching can be computed in polynomial time.

5.2 Show that the size of a maximal matching in a graph G is a lower bound on the size of the minimum vertex cover in G.

5.3 Give a 2-approximation algorithm for cardinality vertex cover.

5.4 What is the relation between the (cardinality) vertex cover problem and the (cardinality) set cover problem?

6 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.

6.1 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.

6.2 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*.

6.3 Give a 3-approximation algorithm for bottleneck TSP (remember to prove that it is a 3-approximation algorithm).

7 Asymmetric TSP Solve exercise 1.3 in [2].