Weekplan: Approximation Algorithms III

Inge Li Gørtz

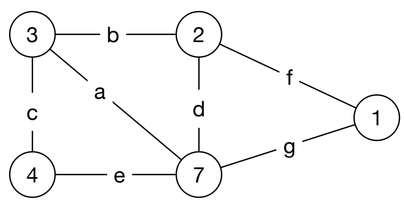
References and Reading

[1] Algorithm Design, Kleinberg and Tardos, Addison-Wesley, section 11.4.

[2] The Design of Approximation Algorithms, Williamson and Shmoys, Cambridge Press, section (proof of Thm 2.4) and 2.3 (proof of Thm. 2.9).

We expect you to read [1] and [2] in detail.

1 [w] **Weighted Vertex Cover** Run the pricing algorithm for weighted vertex cover on the graph below. Consider the edges in alphabetical order.



2 Tight Example for Weighted Vertex Cover Give a tight example for the pricing algorithm for weighted vertex cover.

3 Weighted Vertex Cover Consider the following pricing algorithm for weighted vertex cover. Repeat until all edges are covered: Simultaneously raise the prices of all uncovered edges until some node goes tight. Add all tight nodes to the vertex cover S.

3.1 Run the algorithm on the example from Exercise 1.

3.2 Relate the algorithm to the greedy set cover algorithm.

3.3 Analyze the approximation factor of the algorithm.

4 Hitting Set Consider the Hitting Set Problem defined as follows. We are given a set $A = a_1, \ldots, a_n$ and a collection $B_1, B_2, \ldots, B_m$ of subsets of A. Each element $a_i \in A$ has a weight $w_i \geq 0$. A set H is a hitting set if $H \cap B_i \neq \emptyset$ for all i. The problem is to find a hitting set $H \subseteq A$ such that the total weight of the elements in H is as small as possible.

4.1 Formulate the vertex cover problem as a hitting set problem.

4.2 Formulate the minimum spanning tree problem as a hitting set problem.

4.3 Let $b = \max_i |B_i|$ denote the maximum size of any of the sets $B_1, B_2, \ldots, B_m$. Use the pricing method to give a polynomial-time approximation algorithm for this problem that finds a hitting set whose total weight is at most b times the minimum possible.

5 Asymmetric TSP Solve exercise 1.3 in [2].

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).