

## Exercises

- 1.1 In the set cover problem, the goal is to find a collection of subsets indexed by  $I$  that minimizes  $\sum_{j \in I} w_j$  such that

$$\left| \bigcup_{j \in I} S_j \right| = |E|.$$

Consider the *partial cover problem*, in which one finds a collection of subsets indexed by  $I$  that minimizes  $\sum_{j \in I} w_j$  such that

$$\left| \bigcup_{j \in I} S_j \right| \geq p|E|,$$

where  $0 < p < 1$  is some constant.

- (a) Give a polynomial-time algorithm to find a solution to the partial cover problem in which the value is no more than  $c(p) \cdot \text{OPT}$ , where  $c(p)$  is a constant that depends on  $p$ , and  $\text{OPT}$  is the value of the optimal solution to the set cover problem.
- (b) Give an  $f(p)$ -approximation algorithm for the partial cover problem, such that  $f$  is non-decreasing in  $p$  and  $f(1) \leq H_{|E|}$ .
- 1.2 In the *directed Steiner tree problem*, we are given as input a directed graph  $G = (V, A)$ , nonnegative costs  $c_{ij} \geq 0$  for arcs  $(i, j) \in A$ , a root vertex  $r \in V$ , and a set of terminals  $T \subseteq V$ . The goal is to find a minimum-cost tree such that for each  $i \in T$  there exists a directed path from  $r$  to  $i$ .
- Prove that for some constant  $c$  there can be no  $c \log |T|$ -approximation algorithm for the directed Steiner tree problem, unless  $P = NP$ .
- 1.3 In the *metric asymmetric traveling salesman problem*, we are given as input a complete directed graph  $G = (V, A)$  with costs  $c_{ij} \geq 0$  for all arcs  $(i, j) \in A$ , such that the arc costs obey the *triangle inequality*: for all  $i, j, k \in V$ , we have that  $c_{ij} + c_{jk} \geq c_{ik}$ . The goal is to find a *tour* of minimum cost, that is, a directed cycle that contains each vertex exactly once, such that the sum of the cost of the arcs in the cycle is minimized.
- One approach to finding an approximation algorithm for this problem is to first find a minimum-cost strongly connected *Eulerian* subgraph of the input graph. A directed graph is strongly connected if for any pair of vertices  $i, j \in V$  there is a path from  $i$  to  $j$  and a path from  $j$  to  $i$ . A directed graph is Eulerian if the indegree of each vertex equals its outdegree. Given a strongly connected Eulerian subgraph of the input to the problem, it is possible to use a technique called "shortcutting" (discussed in Section 2.4) to turn this into a tour of no greater cost by using the triangle inequality.
- One way to find a strongly connected Eulerian subgraph is as follows: We first find a *minimum mean-cost cycle* in the graph. A minimum mean-cost cycle is a directed cycle that minimizes the ratio of the cost of the arcs in the cycle to the number of arcs in the cycle. Such a cycle can be found in polynomial time. We then choose one vertex of the cycle arbitrarily, remove all other vertices of the cycle from the graph, and repeat. We do this until only one vertex of the graph is left. Consider the subgraph consisting of all the arcs from all the cycles found.
- (a) Prove that the subgraph found by the algorithm is a strongly connected Eulerian subgraph of the input graph.
- (b) Prove that the cost of this subgraph is at most  $2H_n \cdot \text{OPT}$ , where  $n = |V|$  and  $\text{OPT}$  is the cost of the optimal tour. Conclude that this algorithm is a  $2H_n$ -approximation algorithm for the metric asymmetric traveling salesman problem.

**1.4** In the *uncapacitated facility location problem*, we have a set of clients  $D$  and a set of facilities  $F$ . For each client  $j \in D$  and facility  $i \in F$ , there is a cost  $c_{ij}$  of assigning client  $j$  to facility  $i$ . Furthermore, there is a cost  $f_i$  associated with each facility  $i \in F$ . The goal of the problem is to choose a subset of facilities  $F' \subseteq F$  so as to minimize the total cost of the facilities in  $F'$  and the cost of assigning each client  $j \in D$  to the nearest facility in  $F'$ . In other words, we wish to find  $F'$  so as to minimize  $\sum_{i \in F'} f_i + \sum_{j \in D} \min_{i \in F'} c_{ij}$ .

- (a) Show that there exists some  $c$  such that there is no  $(c \ln |D|)$ -approximation algorithm for the uncapacitated facility location problem unless  $P = NP$ .
- (b) Give an  $O(\ln |D|)$ -approximation algorithm for the uncapacitated facility location problem.

**1.5** Consider the vertex cover problem.

- (a) Prove that any extreme point of the linear program

$$\begin{aligned} & \text{minimize } \sum_{i \in V} w_i x_i \\ & \text{subject to } x_i + x_j \geq 1, \quad \forall (i, j) \in E, \\ & \quad \quad \quad x_i \geq 0, \quad i \in V. \end{aligned}$$

has the property that  $x_i \in \{0, \frac{1}{2}, 1\}$  for all  $i \in V$ . (Recall that an *extreme point*  $x$  is a feasible solution that cannot be expressed as  $\lambda x^1 + (1 - \lambda)x^2$  for  $0 < \lambda < 1$  and feasible solutions  $x^1$  and  $x^2$  distinct from  $x$ .)

- (b) Give a  $\frac{3}{2}$ -approximation algorithm for the vertex cover problem when the input graph is planar. You may use the facts that polynomial-time LP solvers return extreme points, and that there is a polynomial-time algorithm to 4-color any planar graph (i.e., the algorithm assigns each vertex one of four colors such that for any edge  $(i, j) \in E$ , vertices  $i$  and  $j$  have been assigned different colors).

**1.6** In the *node-weighted Steiner tree problem*, we are given as input an undirected graph  $G = (V, E)$ , node weights  $w_i \geq 0$  for all  $i \in V$ , edge costs  $c_e \geq 0$  for all  $e \in E$ , and a set of terminals  $T \subseteq V$ . The cost of a tree is the sum of the weights of the nodes plus the sum of the costs of the edges in the tree. The goal of the problem is to find a minimum-weight tree that spans all the terminals in  $T$ .

- (a) Show that there exists some  $c$  such that there is no  $(c \ln |T|)$ -approximation algorithm for the node-weighted Steiner tree problem unless  $P = NP$ .
- (b) Give a greedy  $O(\ln |T|)$ -approximation algorithm for the node-weighted Steiner tree problem.