Approximation Algorithms (236521) - HW1

Submission deadline: 26.11.18 (Monday)

Guidelines:

• The homework should be submitted individually.

• You are required to provide complete and formal proofs of your claims.

• The homework should be submitted to Ariel Kulik’s mailbox (on the 5th floor of Taub building).

1. Design a 2-approximation algorithm for the problem of finding a minimum cardinality maximal matching in an undirected graph.

2. In the knapsack problem we are given a set of \( n \) elements \( I \), where each element \( i \in I \) is associated with a non-negative integral weight \( w_i \) and a non-negative integral profit \( p_i \). We are also given a budget \( B \in \mathbb{N} \). The objective is to find a subset of elements \( S \subseteq I \) such that \( \sum_{i \in S} w_i \leq B \), and \( \sum_{i \in S} p_i \) is maximal.

Let \( k > 0 \) be some fixed integer. Consider Algorithm 1:

Algorithm 1: Knapsack

1. Set \( R = \emptyset \).
2. Sort the elements in \( I \) by their profit to weight ratio. That is, we obtain \( I = \{e_1, e_2, \ldots, e_n\} \) such that for every \( i < j \) we have \( \frac{p_i}{w_i} \geq \frac{p_j}{w_j} \).
3. for every \( S \subseteq I, |S| \leq k, \sum_{i \in S} w_i \leq B \) do
   4. Set \( T = S \)
   5. for \( i = 1 \) to \( n \) do
      6. Let \( T' = T \cup \{e_i\} \). If \( \sum_{j \in T'} w_j \leq B \) set \( T = T' \)
   7. end for
   8. If \( \sum_{i \in T} p_i > \sum_{i \in R} p_i \) then set \( R = T \)
5. end for
10. Return \( R \)

(a) Show that the algorithm is polynomial in the input size.

(b) Show that the algorithm is \( (1 - \frac{1}{k+1}) \)-approximation for knapsack.

Hint: Consider the iteration in which \( S \) is the set of \( k \) elements with highest profits in an optimal solution.

3. In the minimum-cost Steiner tree problem we are given an undirected, complete graph \( G = (V, E) \) with nonnegative costs \( c_{ij} \) for all edges \( (i, j) \in E \). The set of vertices is partitioned into terminals \( R \) and non-terminals \( V \setminus R \). The goal is to find a minimum-cost tree containing all the terminals.
(a) Suppose initially that the edge costs obey the triangle inequality, i.e., $c_{ij} \leq c_{ik} + c_{kj}$ for all $i, j, k \in V$. Consider computing a minimum spanning tree of $G[R]$ ($G[R]$ is the subgraph of $G$ induced by the terminals $R$). Prove that this gives a 2-approximation algorithm for the minimum-cost Steiner tree problem.

(b) Now suppose that edge costs do not obey the triangle inequality, and that the input graph $G$ is connected but not necessarily complete. Let $\tilde{c}_{ij}$ be the distance of the shortest path from $i$ to $j$ in $G$. Consider running the algorithm above on the complete graph $\tilde{G}$ on $V$ with edge costs $\tilde{c}_{ij}$, to obtain a tree $\tilde{T}$. To compute a tree $T$ in $G$, for each edge $(i, j)$ in $\tilde{T}$, we add to $T$ all edges in a shortest path from $i$ to $j$ in $G$. If the resulting graph $T$ has a cycle, arbitrarily remove a single edge from the cycle, and repeat the process until $T$ is acyclic. Show that this is still a 2-approximation algorithm for the minimum-cost Steiner tree problem on the original (incomplete) input graph $G$.

4. In class we saw an $H_n$-factor greedy algorithm for the minimum weighted set cover problem.

(a) Give a tight example that admits $w(I) = H_n \cdot \text{OPT} - \epsilon$ for all $\epsilon > 0$ ($I$ is the cover that the greedy algorithm shown in class outputs).

(b) Consider the following variant of (unweighted) set cover: Given a set of elements $E$, $m$ subsets of the elements $S_1, S_2, \ldots, S_m \subseteq E$ (such that $\bigcup S_j = E$) and an integer $k \leq m$, select $k$ subsets such that their union has the maximum cardinality.

Give a $(1 - \frac{1}{e})$-approximation algorithm for this problem.

(c) Consider the following problem: Given a set of elements $E$, and $m$ subsets of the elements $S_1, S_2, \ldots, S_m \subseteq E$ each with weight $w_j \geq 0$, find a subset $S \subseteq E$, $|S| = k$ such that we maximize the total weight of the subsets $S_j$ such that $S \cap S_j \neq \emptyset$. Give a $(1 - \frac{1}{e})$-approximation algorithm for this problem.

$\tilde{G}$ is sometimes called the metric completion of $G$. 