Introduction to approximation and randomized algorithms

1st exercise session 16th October 2019

Exercise 1. Formulate the following problems as integer or linear programs:

1. Knapsack problem: On input we get n items with weights $w_1, \ldots, w_n$ and costs $c_1, \ldots, c_n$ and also the size of the knapsack W. We want to find a subset of items of maximal cost such that they fit into the knapsack.

2. Minimum vertex cover: On input we get an undirected graph $G = (V,E)$. We want to find a set $U \subseteq V$ of minimal size such that each edge from E is incident with at least one vertex from U.

3. Maximum cut (MAXCUT): On input we get an undirected graph $G = (V,E)$ with weights w_e on edges. We want to partition vertices in V into two subsets A and B such that the sum of weights of edges going between A and B is maximal.

4. 3-coloring: On input we get an undirected graph $G = (V,E)$. We want to construct an integer program such that it has a feasible solution if and only if G is 3-colorable.

5. Maximum flow: On input we get an oriented graph $G = (V,E)$ with edge capacities w_e and special vertices s and t. We want to find the maximum flow from s to t with respect to the capacities w_e.

6. Minimal cut: On input we get an oriented graph $G = (V,E)$ with edge weights w_e and special vertices s and t. We want to find a set $F \subseteq E$ of minimum weight such that the graph $G \setminus E$ doesn't contain a path from s to t. Alternatively this problem can be defined as finding a partitioning of V

into A and B such that $s \in A$, $t \in B$ and the sum of weights of edges going from A to B is minimal.

Exercise 2. Let us have a problem MAXIMUM PLANAR SUBGRAPH. On input we get an undirected graph $G = (V,E)$ and we want to find a set $F \subseteq E$ such that the graph $G' = (V,F)$ is planar. Our goal is to maximize $|F|$. Try to construct a 3-approximation algorithm for this problem.

In this part we will look at the problem of minimal vertex cover. You can find its definition in exercise 1.

Exercise 3. Suppose we have a greedy algorithm which repeats this step: If there is at least one uncovered edge then take one of its end vertices and add it to the vertex cover. Does there exist a $c > 1$ such that this algorithm is c-approximation for minimum vertex cover?

Exercise 4. We'll modify the algorithm from the previous exercise in the following way: Whenever there exists an uncovered edge, then we add to the vertex cover such a vertex that covers the highest number of currently uncovered edges. Does there exist a $c > 1$ such that this algorithm is c-approximation for minimum vertex cover?

Exercise 5. Let us have yet another greedy algorithm for vertex cover. Now consider an algorithm that repeats the following step: If there is at least one uncovered edge then take both of its end vertices and add them to the vertex cover. Does there exist a $c > 1$ such that this algorithm is c-approximation for minimum vertex cover?

Exercise 6. One of the ways of constructing an approximation algorithms is rounding of linear relaxation. Look at the integer program from exercise one and make its linear relaxation. Suppose that we have an optimal solution for the relaxation. With that optimal solution, are you able to construct a 2-approximation algorithm?