Spring 2018, CS 583: Approximation Algorithms

Homework 1

Due: 02/08/2018

Instructions and Policy: Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people. You may be able to find solutions to the problems in various papers and books but it would defeat the purpose of learning if copy them. You should cite all sources that you use.

Read through all the problems and think about them and how they relate to what we covered in the lectures. Solve as many problems as you can. Please submit your solutions to at least 4 problems.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary. Your job is to convince me that you know the solution, as quickly as possible.

Problem 1 The Greedy algorithm for Max-k-Coverage and Set Cover can be implemented even if the sets in the set system are implicitly defined as long as we have the following oracle: given a subset $\mathcal{U}' \subseteq \mathcal{U}$ of the remaining uncovered elements from $\mathcal{U}$, return the set in the set system that covers the maximum number of elements from $\mathcal{U}'$. Some times we only have an *approximate* oracle. Suppose we only have an α-approximate oracle for some $\alpha \leq 1$ which outputs a set from the set system that has at least α times the number of elements covered by the best set. Show that with an α-approximate oracle Greedy gives a $(1 - e^{-\alpha})$-approximation. Note that this better than the easier bound of $\alpha(1 - 1/e)$.

Problem 2 We saw a randomized rounding algorithm for Set Cover that converts a fractional solution x to the LP relaxation to a feasible solution. The analysis was not so clean because we need to worry both about the cost as well as the feasibility. In this problem we will analyze a small variant that has several advantages. It also highlights the idea of *alteration* in probabilistic algorithms and analysis. In the following we will use Δ to denote the cardinality of the largest set and k to be the maximum frequency of any element.

The algorithm takes a fractional solution x^* and rounds in two steps. In the first step for each set j the algorithm picks S_j independently with probability $\min\{1, \alpha x_j^*\}$ for some parameter $\alpha \geq 1$ that we will pick later. Let J_1 be the indices of sets chosen in this step. Let R be the set of elements that are uncovered by J_1. For each element $i \in R$, the algorithm picks the cheapest set j_i (ties broken arbitarily) that covers i and includes it J_2. Note that the second step is deterministic conditioned on J_1, and in the worst case we may pick n sets in this second step. The algorithm outputs $J = J_1 \cup J_2$, and by construction J is a feasible set cover. It remains to bound the expected cost of J.

- Suppose α is chosen to be $\ln \Delta$. Obtain an upper bound on $\Pr[i \in R]$.
- For $i \in [1..n]$, let j_i be the index of the cheapest set that covers element i, and let $W = \sum_{i \in [n]} w_{j_i}$. Prove that $W \leq \Delta \cdot \sum_j w_j x_j$ for any feasible fractional solution x. In particular $W \leq \Delta \cdot OPT_{LP}$.
- Combine the preceding two to show that the expected cost of the set cover J is atmost $(1 + \ln \Delta) OPT_{LP}$.

Extra Credit: By choosing $\alpha = k(1 - e^{\frac{-\ln \Delta}{k-1}})$, show that the expected cost is at most $(1+(k-1)(1-e^{-\frac{\ln \Delta}{k-1}})) \cdot OPT_{LP}$. Prove that this bound is always at most $\beta = \min\{k, 1+\ln \Delta\}$, and is in fact a constant factor smaller than k when $k = 1 + \ln \Delta$.

Problem 3 Problem 13.4 from Vazirani book.

Problem 4 Problem 1.4 from Shmoys-Williamson book.

Problem 5 Problem 3.6 from Williamson-Shmoys book.

Problem 6 The multiple knapsack problem (MKP) is the following. Like in the standard knapsack problem the input consists of n items, each of which has a profit p_i and a size s_i. However, we are now given m knapsacks with capacities $B_1, B_2, \ldots, B_k$.

- Describe a pseudo-polynomial time exact algorithm for the problem when $k = 2$.
- Prove that even for $k = 2$ and unit profits the problem is NP-Hard. Also prove that there is no FPTAS for the same setting. (Hint: use a reduction from the Partition problem.)

Problem 7 Consider the MKP problem as in the previous problem. Consider a Greedy algorithm that picks each knapsack in turn and packs it using a α-approximate algorithm for the single knapsack problem over the remaining items.

- Prove that if all knapsacks have the same capacity then you obtain a $(1 - e^{-\alpha})$-approximation by showing that the Greedy algorithm can be interpreted as solving a Max-k-Cover problem on an implicit set system.
- **Extra Credit:** Assume $\alpha = 1$. Show that the Greedy algorithm gives a 1/2-approximation even if the capacities are non-uniform and the order of the knapsacks is chosen arbitrarily.