Fall 2021, CS 583: Approximation Algorithms

Homework 1

Due: 09/16/2021 in Gradescope

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 and write in your own words.

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 Problems 1, 4, and at least 2 other problems. Some problems are closely related so it may benefit you to solve them together or view them as parts of an extended problem.

Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.

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 bound is better than the easier bound of $\alpha(1-1/e)$.

Problem 2 We saw a randomized rounding algorithm with alteration for Set Cover that converts a fractional solution x to the LP relaxation to a feasible solution with expected cost at most $(1+\ln\Delta)\text{OPT}_{LP}$ where Δ is the maximum set size. We also saw a simple deterministic rounding algorithm that yields an f-approximation where f is the maximum frequency (recall that $f = 2$ for Vertex Cover). Can we obtain the best of these two algorithms? By adapting the alteration algorithm suitably show that one can obtain a solution with expected cost at most $(1 + (f-1)(1 - e^{-\frac{\ln\Delta}{f-1}})) \cdot \text{OPT}_{LP}$. Prove that this bound is always at most $\beta = \min\{f, 1+\ln\Delta\}$, and is in fact a constant factor smaller than f when $f = 1+\ln\Delta$; this bound is better than choosing the better of the two previous algorithms.

Problem 3 Consider a budgeted version of the maximum coverage problem. We are given m sets $S_1, S_2, \ldots, S_m$, each a subset of a set $\mathcal{U}$. Each set S_i has a non-negative cost c_i and we are also given a budget B. The goal is to pick sets of total cost at most B so as to maximize the number of elements covered.

- Show that if $c_i \leq \epsilon B$ for $1 \leq i \leq m$ then the Greedy algorithm yields a $1 - 1/e - \epsilon$ approximation (assume ϵ is sufficiently small if necessary).
- For any fixed $\epsilon > 0$ obtain a $1 - 1/e - \epsilon$ approximation.

Problem 4 Problem 13.4 from Vazirani book. Do any two of the three subquestions. Extra credit for all three.

Problem 5 Problem 1.4 from Williamson-Shmoys book.

Problem 6 Problem 3.6 from Williamson-Shmoys book.

Problem 7 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.

Problem 8 Consider the MKP problem as in the previous problem but with uniform capacities. 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 Greedy yields a a $(1 - e^{-\alpha})$-approximation.
- **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.

Problem 9 Suppose we are given n points $p_1, p_2, \ldots, p_n$ in the Euclidean plane. We wish to cover the points with as few unit radius disks as possible. Note that the disks can be positioned anywhere in the plane.

- How would you implement the Greedy algorithm for this Set Cover instance in polynomial time?
- Describe a constant factor approximation for this covering problem. It may be easier to first think of the case where you want to cover with squares with unit side length.

Problem 10 Knapsack Cover is the following problem. We are given n items where each item i has a non-negative size s_i and a non-negative cost c_i, and a knapsack of capacity B. The goal is to choose a minimum cost subset of items such that their total size is at least B.

- Describe a constant factor approximation for this covering version of Knapsack.
- **Extra Credit:** Obtain an FPTAS for this problem.