Fall 2021, CS 583: Approximation Algorithms

Homework 3

Due: 10/19/2021 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 solutions to Problem 1 and at least three 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 We have seen the generalized assignment problem (GAP). There is a maximization version called Max-GAP — see the notes for the formal definition which also sketches a 2-approximation via the natural LP. Here we consider a strengthened LP formulatio called the configuration LP. For each machine j let $\mathcal{S}_j = \{S \subseteq J \mid \sum_{i \in S} p_{ij} \leq c_j\}$ denote the set of all feasible subsets of items that fit in the bin j. We need to pick for each bin j a feasible subset. Further, an item can only be packed in one bin. Based on this we can write the following LP with an exponential number of variables. For each $S \in \mathcal{S}_j$ and j we have a variable $x(S, j)$ to indicate if we choose to pack bin j with itemset S. Let $w(S, j) = \sum_{i \in S} w_{i,j}$. Then we can write the following LP relaxation.

$$\max \sum_{j=1}^{m} \sum_{S \in \mathcal{S}_j} w(S,j) x(S,j)$$

$$\begin{aligned} \sum_{S \in \mathcal{S}_j} x(S,j) &= 1 \quad j \in [m] \\ \sum_{j=1}^{m} \sum_{S \in \mathcal{S}_j : i \in S} x(S,j) &\leq 1 \quad i \in [n] \\ x(S,j) &\geq 0 \quad j \in [m], S \in \mathcal{S}_j \end{aligned}$$

- Note that the LP relaxation has only $(m + n)$ non-trivial constraints and hence we are guaranteed an optimum solution with polynomial support. Suppose we are given a feasible feasible fractional solution of polynomial size. Describe a randomized rounding algorithm to obtain a $(1 - 1/e)$-approximation with respect to the value of the LP solution.

- Write the dual of the LP solution. Show that the separation oracle of the dual corresponds to the Knapsack problem. (Although the separation oracle is NP-Hard, one can use the FPTAS to obtain an overall $(1 - \epsilon)$-approximation for the above LP in polynomial time for any fixed $\epsilon > 0$.)

Problem 2 We have seen parallel machine scheduling to minimize the maximum load. Greedy list scheduling gives a 2-approximation while ordering the jobs in decreasing order of size gives a 4/3-approximation. Here we consider a more general version which models resources on a machine. Suppose each machine has d-resources (such as CPU, memory, disk). For each job J_i we associate a non-negative d-deminsional vector $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,d})$ where $v_{i,k}$ is the load that J_i places on resource k. The goal is to assign the vectors/jobs to m identical machines to minimize the maximum load. More formally the vector load on a machine is the vector sum of the jobs assigned to it and the load of the machine is ℓ_∞ norm of the load vector.

- Describe a simple greedy algorithm and prove that it yields a $(d + 1)$-approximation. Any other approach is also fine (including LP based algorithms).
- **Extra Credit:** Obtain an $O(\ln d)$-approximation.

Problem 3 Problem 2.10 from Williamson-Shmoys book.

Extra Credit: For the same problem as in the preceding one consider a local-search algorithm that start with an arbitrary set $S \subseteq E$ of k elements and does a swap if it improves the value. Prove that this gives a 1/2-approximation. *Hint: Set up an (arbitrary) matching between a local optimum S and an optimum solution S^* and consider the swaps corresponding to this matching.*

Problem 4 Problem 2.1 from Williamson-Shmoys book.

Problem 5 The k-Center problem can be alternatively thought of as the following. Given a finite metric space (V, d) and a radius parameter $R > 0$, what is the smallest $\alpha \geq 1$ such that the points in V can be covered by k balls of radius αR where each ball is centered at a point in V? In *Priority* k-Center we are given (V, d), k, and for each $v \in V$ a radius $r(v) > 0$. The goal is to find smallest α such that there are k centers $C = \{c_1, c_2, \ldots, c_k\}$ with the property that $d(v, C) \leq \alpha r(v)$ for each $v \in V$.

- Describe a greedy style 2-approximation for this problem.

- Write a feasibility LP for the smallest α and show that one can obtain a 2-approximation with respect to the lower bound given by the LP relaxation.

Problem 6 Problem 2.14 from Williamson-Shmoys book. This is mainly to introduce you to the edge disjoint paths problem which is connected to the congestion minimization problem we saw. There is a large amount of work on this topic.

Problem 7 Problem 1.4 from Williamson-Shmoys book. This was also in HW 1. You should solve this only if you have not submitted it earlier.

Problem 8 Problem 5.6 from Shmoys-Williamson book.