Spring 2018, CS 583: Approximation Algorithms

Homework 3

Due: 03/08/2018

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 In the Rectangle Independent Set problem (RIS) we are given a collection of n axis-parallel rectangles $\mathcal{R} = \{R_1, \ldots, R_n\}$ in the plane and the goal is to find a maximum cardinality subset of the rectangles that do not overlap. In this problem you will derive an $\Omega(1/\log n)$-approximation. Given a horizontal line L let $\mathcal{R}_a$ be the set of rectangles in $\mathcal{R}$ that lie above L, and let $\mathcal{R}_b$ be the set of rectangles that lie below L, and let $\mathcal{R}_c$ be the set of rectangles that intersect L.

- Describe a polynomial-time algorithm that finds an optimum solution for rectangles in $\mathcal{R}_c$.
- Prove that, given $\mathcal{R}$, there is a line L that can be found in polynomial-time such that $|\mathcal{R}_a|$ and $|\mathcal{R}_b|$ are both at most $\lceil n/2 \rceil$.
- Use the above two parts to design a divide and conquer style algorithm that achieves the desired $\Omega(1/\log n)$-approximation.
- Does the algorithm extend to the weighted case?

Note: Recently a quasi-polynomial time approximation scheme has been found for this problem. For the unweighted case there is an $O(\log \log n)$ approximation and for the weighted case an $O(\log n / \log \log n)$ approximation. Finding a PTAS or a constant factor approximation are major open problems. You can write a natural LP relaxation for this problem that extends the one for intervals. It is an open problem whether this LP relaxation has a constant factor integrality gap. You can convert the argument in the problem into an algorithm that shows that the integrality gap is at most $O(\log n)$. The $O(\log n/\log\log n)$ approximation is also with respect to the LP.

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.

Problem 3 Problem 2.1 from Williamson-Shmoys book.

Problem 4 Problem 2.10 from Williamson-Shmoys book.

Problem 5 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 6 Problem 9.2 from Shmoys-Williamson book. However, there is a slight typo in the data given for the problem. Assume, that for all facilities, the opening cost is 2. The distances c_{ij} remain as given in the problem.

Problem 7 Problem 5.6 from Shmoys-Williamson book.